(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 March 2000 (02.03.2000)

PCT

(10) International Publication Number
WO 00/11590 A1

(51) International Patent Classification[7]: G06F 17/60, G01S 13/02

(21) International Application Number: PCT/US99/18969

(22) International Filing Date: 20 August 1999 (20.08.1999)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/097,370 21 August 1998 (21.08.1998) US
09/339,740 24 June 1999 (24.06.1999) US

(71) Applicant: PINPOINT CORPORATION [US/US]; One Fortune Drive, Billerica, MA 01821 (US).

(72) Inventors: UNDERRINER, Kevin; 39 Pickpocket Road, Exeter, New Hampshire (US). JOHNSON, Charles, L.; 28 Concolor Avenue, Newton, Massachusetts 02458-2524 (US). LEVY, Samuel; 34 High Street, Newton, Massachusetts 02461-1939 (US).

(74) Agent: ANASTASI, John, N.; Wolf, Greenfield & Sacks, P.C., Federal Reserve Plaza, 600 Atlantic Avenue, Boston, MA 02210 (US).

(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW.

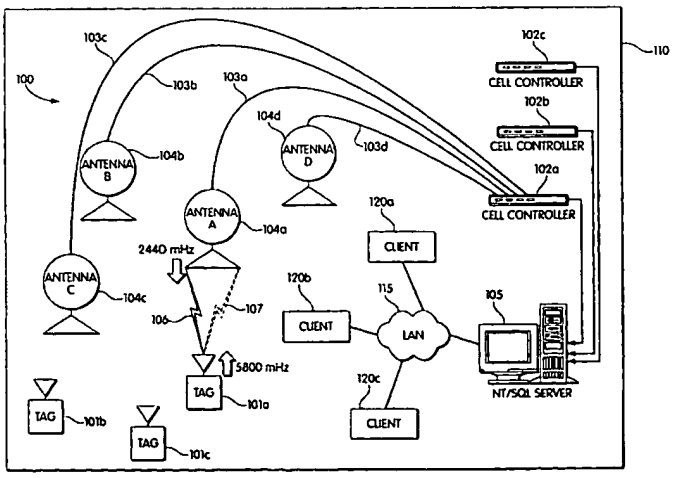
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published:**
— *with international search report*

(48) Date of publication of this corrected version:
16 August 2001

*[Continued on next page]*

(54) Title: METHOD AND APPARATUS FOR TRANSMITTING DATA IN A TRACKING SYSTEM




(57) Abstract: System for tracking mobile tags. Cell controllers with multiple antenna modules generate a carrier signal which is received by the tags. Tags shift the frequency of the carrier signal, modulate an identification code onto it, and transmit the resulting tag signal at randomized intervals. The antennas receive and process the response, and determine the presence of the tags by proximity and triangulation. The recursive-least squares (RLS) technique is used in filtering received signals. Distance of a tag from an antenna is calculated by measuring the round trip signal time. The cell controllers send data from the antenna to a host computer. The host computer collects the data and resolves them into positional estimates. Data are archived in a data warehouse, such as an SQL Server. A software interface is described as part of the system for use by various subscribers and publishers of data as initially generated by cell controllers.

WO 00/11590 A1

(15) **Information about Correction:**
see PCT Gazette No. 33/2001 of 16 August 2001, Section II

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

# METHOD AND APPARATUS FOR TRANSMITTING DATA IN A TRACKING SYSTEM

## Background of the Invention

The invention relates to tracking systems and, more particularly, to systems designed to track articles and personnel.

Various techniques may be used to track articles and personnel. Global positioning systems and local positioning systems are examples that may be used depending on the particular items or persons being located. Included as part of these systems are various embodiments of hardware and software. The hardware may be used, for example, to collect data about the various entities being tracked or located. The hardware may be used in conjunction with the software to facilitate collecting and processing location information.

It is desirable to have an efficient and flexible location system that may be used in a variety of applications that includes hardware and software components.

## Summary of the Invention

In accordance with one aspect of the invention is a location system that includes cell controllers, a server and a client. The cell controllers gather data transmitted by a tag. The server receives and process the data making it available for use by a client. The client requests data from the server. Tags may be associated with objects and persons. The client may be a client software application program or a database software application.

In accordance with another aspect of the invention is a method executed in a computer system for publishing data in a location system to a subscriber. A data publisher registers with a broker. The broker adds the data publisher to a list. A subscriber sends a subscription request to the broker. The broker examines the list to determine if there is a publisher to provide the subscription request. The broker sends a message to the subscriber. The message includes a network location of the data publisher. The subscriber directly connects to the data publisher at the network location.

In accordance with another aspect of the invention is a method executed in a computer system for publishing data in a location system to a subscriber. A subscriber sends a subscription request to the broker. The broker determines if there is a publisher to provide the subscription request. The broker adds the subscriber to a pending list upon determining that there is no publisher to provide the subscription request. A publisher registers with the broker. The broker determines that the publisher publishes data of the subscription request. The broker sends a message to the subscriber including a network address of the publisher. The subscriber establishes a communication channel with the publisher at the network address.

Thus, there is provided an efficient and flexible location system that may be used in a variety of applications that includes hardware and software components.

Brief Description of the Drawings

The above-mentioned and other features of the invention will now become apparent by reference to the following description taken in connection with the accompanying drawings, in which:

Figure 1 shows an overview of a system configured according to the invention;

Figure 2 shows several cell controllers deployed in a multi-story building;

Figure 3 is a block diagram of a tag RF design according to the invention;

Figure 4 is a block diagram of an alternative embodiment of a tag;

Figures 5A-5G are diagrams of a signal as it passes through various stages of the system;

Figure 6 is a block diagram of the cell controller RF design;

Figure 7 is a block diagram of a cell controller active antenna module;

Figure 8 is a block diagram of a modulator RF design;

Figure 9 is a block diagram of a cell controller cable extender module;

Figure 10 is a block diagram of a cell controller;

Figure 11 illustrates extraction of tag data from a series of correlations;

Figures 12A-C are diagrams of tag datagrams;

Figure 13 shows a tag incorporating a delay element;

Figure 14 shows several cell controller receive chains operating in parallel;

Figure 15A is a block diagram of an embodiment of the signal processing hardware of Figure 10;

Figure 15B is a flowchart depicting method steps of one embodiment of the Signal processing hardware unit of Figure 10;

Figure 16 is a block diagram of an embodiment of a signal filtering technique;

Figure 17 is a block diagram of an embodiment of the transversal filter of Figure 16;

Figure 18 is a flowchart depicting method steps of one embodiment of the recursive-least squares (RLS) technique as used in a method step of Figure 15B;

Figures 19A-19E are diagrams of sample waveforms in embodiments of the system of Figure 1;

Figure 20 is a flowchart depicting method steps of one embodiment of approximating a peak of the filtered tag signal;

Figure 21 is an example of an embodiment of a block diagram of a cell controller;

Figure 22 is an example of an embodiment of a block diagram of a cell controller active antenna radio frequency block diagram;

Figure 23 is an example of an embodiment of a block diagram of a tag;

Figure 24 is an example of an embodiment of tag specifications;

Figure 25 is an example of an embodiment of a system in accordance with principles of the invention;

Figures 26A-C are examples of embodiments of a system using tag data as may be gathered by cell controllers;

Figure 27 is an example of an embodiment of various services that may be included in a location system;

Figure 28 is an example of an embodiment of five components that may included in a publish/subscribe technique;

Figures 29-36 demonstrate an example of a publisher/subscriber communication in one embodiment;

Figure 37 is a flowchart including method steps of an embodiment of establishing a publisher/subscriber communication;

Figures 38-46 demonstrate another example of a publisher/subscriber communication in an embodiment;

Figure 47 is a flowchart including method steps in one embodiment for establishing a publisher/subscriber communication;

Figure 48 is an example of an embodiment of the publisher responsibilities;

Figure 49 is an example of an embodiment of the client or subscriber responsibilities;

Figure 50 is an example of an embodiment of a first network topology;

Figure 51 is an example of an embodiment of a second network topology;

Figure 52 is an example of an embodiment of a third network topology;

Figure 53 is an example of an embodiment of a fourth network topology;

Figure 54 is an example of an embodiment of a fifth network topology;

Figure 55 is an example of an embodiment of the unicast mechanism;

Figure 56 is an example of an embodiment of various hardware components included in a location system;

Figure 57 is an example of an embodiment of some hardware and software components that may be included in a location system;

Figure 58 is an example of an embodiment of a flowchart of data types included in a location system;

Figure 59 is an example of an embodiment of various applications and the data used;

Figure 60 is an example of an embodiment of tag antenna distance (TAD) data format;

Figure 61 is an example of an embodiment of location (LOC) data format;

Figure 62 is an example of an embodiment of alert (ALR) data format;

Figure 63 is an example of an embodiment of TSD data format;

Figure 64 is an example of an embodiment of a user interface displaying tags and tag groups;

Figure 65 is an example of an embodiment of a location tree;

Figure 66 is an example of an embodiment of a user interface displaying location rules;

Figure 67 is an example of an embodiment of a format for a location rule;

Figure 68 is an example of an embodiment of a user interface displaying alter rule sets;

Figure 69 is an example of an embodiment of a display of an alert table;

Figure 70 is an example of an embodiment of a user interface display of a location viewer;

Figure 71 is an example of an embodiment of a Visual Basic screen;

Figure 72 is an example of an embodiment of a form for a publisher;

Figure 73 is an example of an embodiment of a code portion that may be associated with a portion of the form from Figure 72;

Figure 74 is an example of an embodiment of a form for a subscriber;

Figure 75 is an example of an embodiment of a subscriber and publisher form;

Figure 76 is an example of an embodiment of a list of subscriber controls; and

Figure 77 is an example of an embodiment of a list of publisher controls.

Detailed Description of Preferred Embodiment(s)

The local positioning system described in paragraphs that follow is an article tracking system that may be used, for example, to track and locate objects indoors. This system may be characterized as a radio-frequency identification system. Generally, such a system may also be used in a variety of other local positioning applications, such as outdoor tracking of objects or locating personnel indoors as well as outdoors. Limitations specific to the embodiments described herein are not meant to imply limitations to the claimed invention.

Radio Frequency Identification (RFID) products typically have three components: (1) a tag (the item being identified), (2) an interrogator (a device which detects the presence

of a tag), and (3) a system (typically including cabling, computers, and software which tie together the tags and interrogators into a useful solution). RFID products are typically designed to detect tags when they pass within range of a few fixed or handheld interrogators.

RFID systems are usually deployed as high-end replacement technology for bar coding. RFID and related systems include passive RFID systems, active RFID systems, infrared ID systems, and Electronic Article Surveillance (EAS) systems.

The tags in a passive RFID system do not carry on-board power. The interrogator in such systems transmits operating power for the tags. Such systems generally have a detection range of a meter or less, although somewhat longer ranges have been achieved. Typically, these systems operate in the 125-kilohertz or 13.56 megahertz radio band.

Most passive RFID systems work as follows. An interrogator emits an electromagnetic field for the purpose of powering the tag. A coil in the tag is powered by the electromagnetic field, causing the tag's circuitry to "wake up." The tag uses this power to send an identifying signal back to the interrogator.

Although some passive RFID systems are read-only (that is, the tags in such system respond to a query by reading information from their memory and sending the information back to the interrogator), the tags used in other passive RFID systems have a limited ability to accept information and instructions from the interrogator, for example read/write capabilities in smart cards (electronic money) and "electronic manifests" in industrial applications.

Passive RFID tags have been employed in conjunction with access control, smart cards, vehicle identification (AVI), waste management, item tracking, animal identification, manufacturing control, materials handling, and a variety of other purposes.

One fundamental design goal of any RFID system is for the weak signal emitted from the tag to be distinguishable from the much stronger signal emitted by the interrogator. Some strategies for doing this include:

Frequency shifting. Circuitry in the tag receives a carrier from the interrogator, translates the signal to another frequency, and emits a response modulated onto that second frequency.

Half duplex operation. The tag is charged by the interrogator. When the interrogator's charging circuit turns off, the tag uses the stored power to respond.

Modulated backscatter. The tag modulates its antenna cross section to identify itself to the interrogator.

Delayed retransmission. Surface acoustic wave (SAW) devices retransmit the interrogator's carrier after a delay. The tag's identity is indicated by time variations in the delayed response.

Active RFID systems require battery-powered tags. The battery permits a longer detection range of between 3 and 100 meters. These systems are capable of locating tags with higher accuracy than passive RFID systems and typically operate in the 400, 900, 2440, or 5800 megahertz bands. Active tags tend to enable multiple tags to be within range of an interrogator by the use of "handshaking" between the tags and interrogator, so that each tag transmits its signal in turn. Communication between tag and interrogator in active RFID systems is also typically faster than with passive tags.

Most active RFID tags respond to the interrogator when polled, in accordance with a communications protocol. Some active RFID tags "chirp" (transmit) a signal spontaneously at predetermined intervals. A tag's chirped signal is detected by the interrogator if the tag is in range of the interrogator.

Infrared systems (IRID), while not RFID systems, also endeavor to detect and identify the position of mobile tags. A typical IRID system includes a tag that chirps its identity at randomized intervals. Infrared readers located in the ceiling detect these transmissions, and report them to a host. The transmission rate from the tag to reader is typically about 600 baud. Motion detectors in the tags enable the tags to transmit more frequently when in motion. The tags are typically about the size of dominos.

EAS systems are intended to deter theft in retail environments. EAS tags are fairly unreliable, very low in cost, and limited in capabilities. Although they track mobile tags, they are not generally considered to be identification products, because EAS tags are uncoded and cannot be distinguished from one another.

Referring to Figure 1, an article tracking system 100 contains the following general components:

Tags: Inexpensive miniature radio frequency transponding tags 101a-c are attached to people and/or items being tracked. Tags 101a-c "wake up" periodically, and "chirp" (transmit) a radio-coded unique identity code (UID). The tags 101a-c are designed so that their range is 50-100 meters in a typical indoor environment, the range mostly being limited by a need to conserve the life of the tag battery, and the requirement that the tag 101a and tag battery be small and thin.

Cell Controllers: Cell controllers 102a-c detect the chirps of tags 101a-c and calculate the distances of those tags 101a-c to active antenna modules 104a-d connected to the cell controllers 102a-c. Each antenna module preferably has a transmit antenna and a receive antenna. In Figure 1, the antenna modules connected to cell controllers 102b and 102c are omitted for simplicity. A cell controller 102a is typically contained in a case and is mounted behind a hung ceiling. The cell controller 102a can receive power from a conventional wall outlet or the equivalent. The cell controller 102a is attached through

coaxial cables 103a-d to the antenna modules 104a-d, respectively, which provide coverage of an area of the indoor facility 110. A tag signal 107, transmitted by a tag 101a, is received by one or more antenna modules 104a-d, and is processed by chips in the cell controller 102a, such as digital signal processing (DSP) chips as may be included in the signal processing hardware. The information resulting from this processing is used to identify both the identity of the transmitting tag 101a and the distance between the tag 101a and each, for example, of the receiving antenna modules 104a-d.

Host Computer: Cell controllers 102a-c are in data communication with a host computer 105, which collects data and information available from the cell controllers 102a-c and archives the data into an open format database, such as an SQL Server.

User Application: In a preferred option, client workstations 120a-c communicate with the host computer 105 over a network, such as a LAN 115. A client application running on each of the client workstations 120a-c can access the SQL Server and presents the data in a way that is useful to an end-user.

The tag 101a does not generate its own radio signal. Rather, an antenna module, for example antenna module 104a, continuously transmits a direct sequence spread spectrum interrogator signal 106 at a first frequency, for example 2440-megahertz. The tag 101a receives this signal 106, modulates its UID code onto the signal 106, and immediately transmits back a frequency-shifted signal 107 at, for example, 5800-megahertz to, for example, antenna 104a. The distance from the antenna module 104a to the tag 101a can then be determined by the cell controller 102a from the round trip time of transmission, accounting for fixed and known delays in the wiring and electronics. The cell controller 102a can quickly switch among antenna modules 104a-d to obtain the distance from the tag 101a to each of antenna modules 104a-d (which receive the return signal 107), and from that information determines the tag's location by triangulation techniques.

The system 100 is designed to be scaleable, allowing addition of cell controllers to existing cell controllers 102a-c and antenna modules to existing antennas modules 104a-d to improve the precision with which tag location is determined. Figure 2 shows how a collection of cell controllers 102a-c can be deployed in the large multistory building 110. As shown in Figure 2, multiple cell controllers 102a-c feed data to a single host computer 105, typically through a TCP/IP communications network. A variety of data protocols and transfer mechanisms can be used in preferred embodiments. For example, if a local area network is not available, connection to the host can be accomplished via RS485, RS232, RS422, power line modem, or a dedicated phone line. Alternatively, specialized modems designed for use on such cables can be employed.

Each of the cell controllers 102a-c can be installed to cover a separate floor 130a-c, respectively, with the exact configuration being modifiable by a system administrator. On floor 130a, cell controller 102a, with its collection of antenna modules 104a-d, is installed in the ceiling 140a. The same configuration of equipment is used on the remaining floors 130b-c. Antenna modules 104a-d are designed to provide good gain downwardly and horizontally, and poor gain upwardly, so that a vertical location (that is, the floor) of a tag 101a can be determined by noting which antenna modules 104a-d receive the strongest signals from the tag 101a. Structurally, a ground plane is placed behind each antenna to reflect signals downward. The horizontal location of a tag 101a is then roughly determined by noting which antenna modules 104a-d receive a strong signal from the tag 101a. The horizontal location of the tag 101a with respect to an antenna module 104a can be determined more precisely by estimating the distance from the tag 101a to each antenna module 104a-d, based on the combined time of flight of the interrogation signal 106 and the tag signal 107. Each "cell," consisting of a cell controller 102a and its antenna modules 104a-d, covers several thousand square feet of floor space. Each cell operates

independently, enabling more cells to be added without affecting the performance of existing cells.

If a user desires to locate tags by "zone," one antenna per zone can be installed. Users wishing to track one or more tags 101a-c moving down hallways, can install antenna modules 104a-d every 20 or so meters along a hallway 130a-c, and calculate the linear location of a tag 101a by measuring the distance from the tag 101a to those antenna modules 104a-d. Customers wishing to triangulate upon the location of a tag 101a may install enough antenna modules such that the tag 101a will be in range of at least two or three of the antenna modules, depending on the coverage of the antenna modules and the geometry of the facility. A typical installation will cover a complete facility 110 with a combination of "zone" and "hallway" coverage at a relatively low cost per square foot, and, over time, upgrade certain areas with enough antenna modules to triangulate upon a tag's position.

Other preferred embodiments may include a wall mounted antenna radiating generally horizontally, rather than vertically. The selection of antenna type may be based on a variety of functional factors familiar to one of ordinary skill in the art.

Tag RF Design

Referring to Figure 3, a tag RF circuitry 300 receives signal 106 at a tag receive antenna 301 and transmits tag signal 107 at a tag transmit antenna 312. The function of the tag RF circuitry 300 is to transpond the incoming spread spectrum signal 106 by frequency translation. The secondary function of the tag RF circuitry 300 is to modulate tag data onto the emitted tag signal 107, under the control of a microprocessor 308. The information emitted on the tag signal 107 includes, in a preferred embodiment of the invention, the

serial number of the tag, datagram headers, and tag data 309 such as that derived from a motion indicator or a low power indicator.

The incoming signal 106 is preferably a direct sequence spread spectrum signal, biphase or quadrature modulated from the cell controller 102a, in the 2440 megahertz band. The signal 106 is received by the tag receive antenna 301, which collects the signal 106 and feeds it into the tag RF circuitry 300.

After the signal 106 is received by the tag receive antenna 301, an Rx (receive) bandpass filter 302 ensures that the tag is receiving signals only in the 2440 megahertz ISM band, rejecting radar signals, electronic newsgathering signals, and the like. In one embodiment, the filter 302 is implemented as an etched coupled stripline filter embedded in the circuit board. The signal 106 is then amplified by an amplifier 303 to ensure that the received signal can be mixed, in a frequency mixer 304, without degrading the signal to noise ratio (SNR).

The frequency mixer 304 translates or shifts the carrier frequency from 2440 megahertz to 5800 megahertz. The incoming signal, with a center frequency of 2440 megahertz, is mixed with the output of a phase locked oscillator (PLO) 305 with a center frequency of 3340 megahertz. This results in a sum frequency of 5800, along with a difference frequency and a variety of harmonics and subharmonics which are removed with a bandpass filter 306. In one embodiment, the PLO 305 consists of a phase locked loop (PLL) chip with three inputs: (1) a sampled output from a voltage controlled oscillator (VCO); (2) a reference tone from a 10 megahertz oscillator; and (3) a frequency programming interface to a microprocessor 308. This generates a pure tone with good phase noise at the 3340-megahertz tag LO frequency. In an alternative embodiment, the PLO 305 outputs a 1670-megahertz tone, which is then doubled to give the desired 3340-megahertz result.

The next element of the tag RF circuitry 300 is a biphase modulator 307 which, under control of the microprocessor 308, can either pass the 5800-megahertz signal unaltered, or change the phase of the signal by 180 degrees. The modulator 307 is implemented as a single pole double throw RF switch 801 that feeds a 180 degree hybrid, as shown in Figure 8. Several forms of modulation can be used, including on-off keyed (OOK) modulation, binary phase-shift keyed (BPSK) modulation, multiple phase-shift keyed (MPK) modulation, and quadrature amplified (QAM) modulation. BPSK is the preferred form of modulation. The output from the modulator 307 is fed into an amplifier 310, then is filtered by a transmitter bandpass filter 311, and the output of filter 311 is emitted from a transmit antenna 312 as the tag signal 107. Since the amplifier 310 operates at high frequency, it consumes significant power, and alternative embodiments (such as that shown in Figure 4) that make this amplifier 310 unnecessary are preferred. The Tx Filter 311, implemented as a 5-pole filter, is necessary to ensure tag compliance with FCC Part 15 requirements.

The tag RF circuitry 300 shown in Figure 3 is intended to illustrate the general functions of a tag 101a-c, with an embodiment that is workable and self-explanatory. Those skilled in the art will be able to combine multiple functions into single elements in order to conserve power and take full advantage of available parts, or implement the same functions with a custom ASIC. Figure 4 shows an alternative embodiment 400 which fulfills the same basic functions as that shown in Figure 3, but with fewer components and using less power. The essential difference between the circuitry 400 shown in Figure 4 and the circuitry 300 shown in Figure 3 is that the modulator 404 in Figure 4 is placed before the frequency mixer 406 in order to reduce the number of components (for example, the amplifier 310 is eliminated) and to conserve power.

In the place of the frequency mixer 304 (Figure 3) or the time delay element 1505 (Figure 13), other signal transmission discriminators may be used to transpond by other methods. For example, a tag, such as the tag 101a, can transpond using backscatter, frequency translation by mixing, frequency translation by taking a harmonic, frequency translation by taking a subharmonic, or by signal delay (such as via a SAW device).

Not shown in Figure 4, but desirable for the tag RF circuitry, is the use of a common crystal reference for both the PLO 407 and clock timing in the microprocessor 405. Accurate timing is an important, if not critical, feature of the system, enabling the cell controllers 102a-c to predict when a tag 101a will transmit a tag signal 107. Using the same crystal reference in the PLO 407 and in the microprocessor 405 clock timing allows the cell controller 102a to accurately calibrate the source by measuring phase shifts in the received signal (as described hereinafter), and to synchronize its clock timing accordingly.

Not shown in Figure 4, but desirable for some applications, is an embodiment where the transmit antenna 409 and receive antenna 401 are combined into a single element, and which uses a diplexer with the single antenna structure.

The manner in which the tags 101a-c are powered depends on the application. (Note that Figure 3 and 4 omit the tag power source.) Typically, a tag 101a will be battery powered, with the RF stage turned on and off under control of the microprocessor 405. In a preferred embodiment, the microprocessor 405 goes into a low power state, where it merely waits until it is time to power up the tag 101a again. In an alternative embodiment, all tag circuitry 400 cycles on and off under analog control, using an RC time constant in the circuit 400 as the timing source.

Using the tag RF circuitry 300 or 400 of Figure 3 or 4, if a tag 101a is in range of two of the cell controllers 102a-c, and those cell controllers are sending pseudonoise with

low cross correlation characteristics, the tag 101a will correctly transpond both signals simultane

Tags 101a-c require a period of time, on the order of a millisecond, to charge up and charge down. During these brief periods, typically, the tags 101a-c will not be stable enough to use, but will nonetheless emit RF into the radio channel through the transmit antenna 409. For high-performance applications, where radio bandwidth is limited, a microprocessor-controlled switch can be added to the tag's transmit chain to eliminate such spurious emissions.

The tag RF circuitry 300, 400 shown in Figure 3 and 4 can be used in conjunction with different pairs of frequencies. The general approach described above works for any two allowable FCC spread spectrum bands. For example, the following combinations are permissible for license-free radio under FCC regulation Part 15.247:

915 megahertz translated to 2440 megahertz.

915 megahertz translated to 5800 megahertz.

2440 megahertz translated to 915 megahertz.

5800 megahertz translated to 915 megahertz.

5800 megahertz translated to 2440 megahertz.

Spread spectrum operation, however, is not required; two licensed narrow bands can be used. However, spread spectrum operation in the 2440 and 5800 megahertz bands is assumed for the remainder of the discussion.

Tag With Time Delay

The tag RF circuitry 300, 400 shown in Figures 3 and 4 use frequency division multiple access, i.e., the tag circuitry 300, 400 receives and emits signals on different frequencies. An alternative embodiment 1500 uses time division multiple access, as shown in Figure 13. For illustrative purposes, assume that the tag circuitry 1500 shown in Figure

13 takes as an input at a receive antenna 1501 a signal at one frequency, such as 915mHz, and emits the same signal through a transmit antenna 1508 at the same frequency after a delay of a microsecond. Assume that a cell controller, such as cell controller 102a, transmits an interrogation signal 106 in bursts every 2 microseconds. A tag, such as tag 101a, takes this signal as an input through the receive antenna 1501. The signal then passes through elements 1502-1504, as in Figures 3 and 4. A time delay element 1505 is then used to delay for a microsecond. The signal then passes through a transmit bandpass filter 1507 and is emitted from the transmit antenna. A SAW device can be used as the time delay element 1505. During the period of the delay, the cell controller ceases transmission, and reflections of the interrogation signal 106 in the environment die down to a minimal level. This half-duplex approach allows single frequency operation, although with lower bandwidth than with a full-duplex frequency shifting approach. As in the frequency-changing tag, the delay-based tag is capable of modulating the response signal by a 180-degree phase shift. In other respects, the tag design 1500 shown in Figure 13 is similar to those shown in Figures 3 and 4.

Cell Controller RF Design

Figure 6 shows the radio stage of a cell controller 102a. The architecture of an antenna module, such as the antenna module 104, is shown in Figure 7. Together, the cell controller 102a and its remote antenna modules 104a-d modulate a baseband square wave input onto a 2440-megahertz carrier, filter the resulting 2440 megahertz signal to comply with FCC transmission requirements, transmit that filtered 2440-megahertz signal through a selected antenna module, receive a returning 5800-megahertz tag response through the same antenna module, extract the I (Inphase) and Q (Quadrature) components of the demodulated baseband signal, and digitize the result for subsequent processing.

Figure 10 shows the main components of the cell controller digital subsystem 650. In summary, the digital subsystem 650 provides a baseband input signal 601, and some number of nanoseconds later receives a demodulated response 107 from a tag 102a. The microprocessor 1001, as noted above, can change the behavior of the radio system by (a) modifying the baseband input signal 601; (b) modifying the chip rate, pseudonoise sequence length, and/or the pseudonoise sequence code; (c) modifying the transmit frequency 610 of radio transmitter 1002 and the receive frequency of radio receiver 1003 within a narrow range; (d) modifying the transmit gain of radio transmitter 1002 and the receive gain of radio receiver 1003; and (e) by switching antenna modules 104a-d.

The demodulated response 107 from the tag 102a is split into I (Inphase) and Q (Quadrature) components by the receiver Radio 1003, and digitized by a digitizer 636. Signal processing hardware 1004, for example a combination of DSP and FPGA components, reduces the output from the digitizer 636, performing correlation operations at high speed. If binary phase-shift keyed (BPSK) modulation is used on the transmitting side, the I and Q channels are correlated separately and combined. For quadrature phase-shift keyed (QPSK) modulation, each channel must be correlated twice, once with each sequence. The correlated data from the signal processing hardware 1004 is processed by a microprocessor 1001, such as a Pentium processor. Communications between the microprocessor 1001 and the host computer 105 is accomplished using a TCP/IP protocol, with Ethernet being preferred.

The data that is input to the transmit chain is a baseband input signal 601 which is a pseudonoise spreading sequence. The length of the sequence and the code encoded in the sequence are set by a cell controller microprocessor 1001, and can be varied depending on signal processing requirements. Thirty-one or 127 bit sequences are typical, giving about 15 dB and 20 dB of compression gain, respectively. The 2440 megahertz and 5780

megahertz bands can support a 40 megahertz baseband input signal 601, and the cell controller 102a is designed to enable this full bandwidth to be utilized.

Figures 5A-5G show an interrogation signal 106 as it passes through various stages of the cell controller RF circuitry 600. Figure 5A shows a square wave baseband input to the modulator 500. Figure 5B shows this baseband input digitally correlated 510. Figure 5C shows an output 520 from a-modulator 602, viewed through a spectrum analyzer centered at 2440 megahertz. Figure 5D shows a spectrum analyzer view 530 of the tag signal 107, centered at 5780 megahertz. Figure 5E shows the demodulated response from tag 107, separated into its I (Inphase) 545 and Q (Quadrature) 540 components. Figure 5F shows the I and Q components, digitally correlated 550. Figure 5G shows the negative of the second derivative of the correlated waveform, combining the I and Q components 560.

The modulator 602 (Figure 6) modulates the baseband input 601 onto a 2440 megahertz carrier. Various forms of modulation are available and well-known to those skilled in the art. For BPSK modulation, the modulator 602 is implemented as a single pole double throw RF switch 801 that feeds a 180-degree hybrid combiner 803, as shown in Figure 8. The modulator 602 is preferably implemented as a QPSK modulator, which duplicates the BPSK modulator with one channel offset by 90 degrees from the other, each channel driven by a different baseband sequence with acceptable cross-correlation characteristics. Higher order modulation is also possible. Modulation by the modulator 602 results in sidelobes extending for hundreds of megahertz, which need to be filtered to comply with FCC requirements. The 2440 megahertz band has an adjacent band, which imposes very strong filtering requirements, best addressed using in the illustrated embodiment using a SAW filter 607 that combines wide passbands with tight stopbands. A wider passband supports a faster chipping rate in the baseband input signal 601, but a narrower passband provides an opportunity to use a wider range of frequency diversity to

avoid jammers and/or support advanced signal processing techniques. The modulator 602 must operate at the same frequency as the available IF filter 607, typically in the range of 200 megahertz to 400 megahertz. A preamplifier 606 is necessary prior to the SAW IF filter 607, and the output of the filter needs to be amplified by an amplifier 608.

A transmit IF oscillator 605, like all other RF oscillators in the cell controller circuitry 600, is phase locked to a 10 megahertz crystal source 603, which is distributed through a filter and splitter network 604 to each of the oscillators. The 10 megahertz source 603 needs to be within a few kilohertz of the 10 megahertz sources on the tags to avoid excessive baseband phase shift.

The output from the IF filter 607 (from amplifier 608) is then mixed by a mixer 609 with the output from a phase locked oscillator (PLO) 611 and is converted to a carrier frequency of 2440 megahertz. The frequency of the PLO 611 can be modified within a narrow range under microprocessor control 610, in order to provide the frequency diversity needed to avoid jammers and/or for various advanced signal processing techniques. The degree of frequency diversity available is related to the specifications of the IF filter 607, with narrower filters permitting a slower chip rate but having more frequency flexibility. Not shown in Figure 6 is a filter typically needed to remove undesirable harmonic and difference frequencies from the output of the mixer 609.

Following the mixer 609 is a driver amplifier 612 which raises the power level of the signal 106, so that it can he driven down the cable 103a to the remote antenna module 104a, and which buffers the output of the mixer 609 for a bandpass filter 613. The RF bandpass filter 613 is needed to remove FCC non-compliant outputs from the mixer 609. A directional coupler 616 provides a port to examine the signal 106 before it is transmitted to the remote antenna modules, for example antenna modules 104a-d.

An attenuator 614 under microprocessor control 615 allows the signal processing software to decrease output power when a tag 101a-d is known to be nearby. This is helpful in circumstances when a nearby tag is known to be over-driven by the cell controller, and/or the signal processing software needs the tag to operate in a more linear range.

The signal is then fed into a diplexer 618, which combines the transmitted 106 and received 107 signals onto a single wire. The diplexer 618 is a highpass/lowpass filter combination 619a that attenuates a received signal 107 relative to the transmit side and attenuates the transmit signal 106 relative to the receive side. Because of the presence of the Tx and Rx bandpass filters 613, 624, the specifications of the diplexer 618 are not very stringent.

The cell controller RF stage 600 shown in Figure 6 supports one remote antenna module 104a-d at a time. In order to support multiple antennas from the same cell controller, the system needs a switch 619, which enables a microprocessor control 620 to rapidly switch from one antenna to the next. The switch 619 takes RF and passes it into one of *n* cables, where *n* is, for example either 8 or 16. The switch 619 also provides DC power to the selected line. The RF signal is coupled into the cables with a capacitor (not shown), which provides DC isolation, and the DC power is coupled into the cables with RF chokes to provide RF isolation. Thus, the DC and RF travel together through a single coaxial cable to the selected antenna.

The rise time of the DC in an antenna is in the range of 20 microseconds, limited by the effective resistance of the circuitry in and characteristics of the antenna and the capacitors needed for operation. To provide antenna switching time in the microsecond range, the DC power to an antenna is preloaded before the RF is switched.

Referring to Figure 7, in an antenna system 700, the combined DC and RF signals arrive through a coaxial cable, such as the cable 103a from the cell controller 102a. A bias-

tee 701 separates the RF signal 710 from the DC signal 712. The DC signal 712 is sent to Tx/Rx power control logic 702 which, in the simplest embodiment is a filter to remove noise from the line and provide a clean 5 volt power source. The RF output 710 from the bias tee 701 is fed into a diplexer 715, which is identical to the diplexer 618 in the cell controller 102a. This is then amplified by an amplifier 703 to the power level allowed by the FCC, and filtered by a filter 704 to remove line and amplifier noise in compliance with FCC regulations. The resulting signal is then sent to a transmit antenna 705.

The transmit antenna 705 and receive antenna 706 are, in this embodiment, wall mounted patch arrays, providing reduced energy in the vertical direction and spreading energy laterally, so that power is not wasted in the floor and ceiling, and so that minimal power is radiated upward. The 5780-megahertz response 107 from the tag 101a is filtered by a filter 707, amplified by an amplifier 708, and sent back down the cable 103a to the cell controller 102a.

The system is designed to use cables 103a-d of a standard length, for example, 20 meters. A cable extender module 900 connects two lengths of cable and supports an extended cable length. Referring to Figure 9, the elements of the module 900 use the DC power 910 from the cable 103 to drive low noise amplifiers 903, 904, which provide enough gain to drive the next section of cable. Bias tees 906, 907 separate the DC power 910 from the RF signals, and diplexers 908, 909 operate to separate the transmit signal 106 from the receive signal 107.

Referring to Figure 6, the signal returning from the antenna module 104a to the cell controller 102a passes through the switch elements 621, 619 and diplexer 618 to the cell controller receive RF chain 622. The signal passes through a combination of a preamplifier 623 and bandpass filter 624, the exact arrangement of which varies based on the parts selected. A digitally controlled receive attenuator 625 under microprocessor control 626 is

used to avoid saturating the receive chain when the tag 101a is known to be nearby. This is necessary to avoid losing the relationship between the I and Q components of the received signal 107, necessary for proper correlation and other signal processing.

The signal then enters an I-Q zero IF demodulator circuitry 627-633. As noted previously, the microprocessor Rx frequency control 635 must be set in tandem with its counterpart in the transmit chain. The resulting signal, an example of which is shown in Figure 5E, is fed into a digitizer 636 (Figure 10) in preparation for digital signal processing.

The embodiment described above is simplified, based on an assumption that the cell controller can send and receive from only one antenna at a given time. Improved performance can be achieved by selecting send and receive antennas independently of each other. Software in the cell controller determines which antenna module receives the best signal from the tag. For example, if a particular tag, such as tag 101a, is close to an antenna, such as antenna 104a, then the antenna 104 will receive a strong signal from the tag 101a. The cell controller 102a can then transmit a signal, such as signal 106, from antenna 104a, and receive the transponded response 107 at antennas 104b, 104c, and 104d in turn. This can result in a stronger signal received at the antennas 104b-d, compared to the signal the antennas 104b-d would have received had the signal 106 been transmitted and the signal 107 received from each of the antenna modules 104b-d independently.

The design 1600 shown in Figure 14 provides multiple receive chains 1610a-1610n operating in parallel. Each of the receive chains 1610a-1610n includes an IQ demodulator, a digitizer, and a correlating element, shown as integer DSPs, for example, integer DSP 1620. Implementing each receive chain on a separate card provides scaleability. The use of multiple receive antenna modules for the same transmitted signal allows the cell controller signal processing software to employ spatial processing techniques to isolate multipath

effects. These techniques take advantage of the fact that the multipath-corrupted response will have different characteristics at each antenna.

Bit Detection

In an ideal environment, a simple triangular correlation peak can be derived from a received tag signal 107, as shown in Figure 5B. Distortions introduced in the radio chain, particularly caused by indoor multipath effects, result in a distorted but nonetheless distinct correlation peak, a function of which is shown in Figure 5G. For the purpose of bit detection, the essential point is to reliably detect the existence of a series of correlations, which indicates the operation of a tag. Figure 11 shows how tag data is extracted from a series of correlations. In the left half of the chart 1110 shown in Figure 11, the tag is transmitting a "one." This is accomplished by setting the tag's modulator 307 to pass the interrogator signal 106 unaltered. When the received tag signal 107 is correlated with the transmitted pseudonoise sequence, essentially identical correlation peaks result. Three such peaks 1120a-c are illustrated here. During the time of the fourth correlation 1120d, the tag flips the phase of the modulator by 180 degrees, indicating a "zero," as shown in the chart 1110 (Zero is encoded as a 180 degree phase change). Since the modulation is changed in the middle of a bit, the fourth correlation data peak 1120d is corrupted, and is best ignored. The fifth and sixth correlation peaks 1120e-f cleanly reflect the 180-degree shift.

Pseudonoise sequences can be varied under microprocessor control at the cell controller. when a tag's presence is first detected, relatively short sequences must be used, as shown in Figure 11. Once the tag's bit timing is ascertained, it is possible to use longer sequences for improved SNR, which is helpful in distance measurement.

An important consideration, not shown in Figure 11, is that the balance between the Inphase and Quadrature components of the received signal will drift over time. This is

because the 10-megahertz sources in the cell controller 102a and the tag 101a will differ typically by a few kilohertz. This factor can be calibrated by noting the phase difference between contiguous correlations, detectable at the baseband by noting changes in Inphase and/or Quadrature components of the received signal. As noted previously, this same calibration process can be used to calibrate the tag clock in reference to the cell controller clock, allowing for precise prediction of tag chirping time, without needing to precisely measure the timing of tag bit transitions.

Interaction Between Cell Controllers and Tags

Each tag is a stand-alone unit that is unaware in any way of the outside world. Each tag has a Unique Identifying Code (UID) associated with the tag when it is manufactured. A tag wakes up periodically and, for a short period of time, converts any incoming 2440-megahertz signal 106 to an outgoing 5780-megahertz signal 107, while modulating its UID and other data onto the outgoing signal 107 which it chirps (transmits). The tag does not communicate with other tags. The tag does not explicitly respond to an interrogation signal, but merely transponds any incoming signal 106 in the 2440-megahertz band, which may or may not include a pseudonoise sequence from a nearby cell controller antenna module 104a. This approach greatly simplifies the design and fabrication of the tag 101a.

Some portion of the time, two or more tags will transpond simultaneously. In many cases, one of the two tags will return a stronger signal than the other tag and some data will be lost in such a collision. To avoid collisions occurring in a repeating pattern, tags "wake up" and chirp their UIDs at randomized times, which can be calculated (by both the tag and the cell controller) based on a pseudorandom number generator which incorporates the tag's UID. For example, for a tag which chirps approximately every 5 seconds, the tag generates pseudorandom numbers between 0.0 and 2.0, and adds these to a 4.0-second minimum

delay time, resulting in a sequence of delay times uniformly distributed between 4.0 and 6.0 seconds.

It is possible to use analog inputs, such as an input from an internal clock or the delay of an RC circuit, to the pseudorandom number generator to vary the seed over time, but a purely digital scheme is preferred to enable the cell controller 102a to accurately forecast a known tag's chirping time. A typical pseudorandom number generator has the form:

$N = rand(seed)$ Formula 1

The resulting N is used as the seed for the next pseudorandom number in the pseudorandom number sequence. When using a pseudorandom number generator of this kind, it is possible that two tags will use the same seed, resulting in their tag signals repetitively colliding. Further, with small differences in tag clocks, all pairs of tags will eventually drift through this synchronized state for some amount of time. To avoid these conditions it is desirable to incorporate each tag's UID, as noted above, into the delay time for that tag, resulting in a different pseudorandom sequence for each tag, that is:

$Delay = f(N,UID)$ Formula 2

One simple example of such a function is:

$Delay = Xor(N, BitRotate (UID, AND(N, 1111_2))$ Formula 3

Referring to Formula 3, it is possible to reconstruct seed from UID, Delay and $And(N,1111_2)$, by calculating $N = Xor(Delay, BitRotate(UID, And(N,1111_2))$.

Referring to Figure 12a, one embodiment of the tag datagram 1400 contains a header 1401 to enable the cell controller to detect the tag's presence, followed by an identifier preamble 1402, followed by the tag's UID 1403. The header 1401 can be of zero length. The identifier preamble 1402 can be implemented, for example, as a validity check such as a cyclic redundancy check (CRC). Given a sufficiently simple Delay function and high clock stability, the cell controller can infer the tag's chirping sequence by noting the timing of a series of chirps of the datagram 1400.

Referring to Figure 12b, in another embodiment of the tag datagram 1410, the tag adds Delay information 1414, thus enabling the cell controller to forecast the transmission time of the tag's next and subsequent chirps of the datagram 1410. In the example of Formula 3, this information would include the data: Delay and And(N,$1111_2$).

Referring to Figure 12c, in another embodiment of the tag datagram 1420, a shorter header is used than in the datagrams 1400, 1410 of Figures 12a and 12b, such that the cell controller is not guaranteed to have enough time to detect the tag's presence before the UID 1423 contained in the tag datagram 1420 is transmitted. Appended to the datagram 1420 is the transmission delay 1425 of the next chirp, enabling the cell controller to anticipate the time that the tag will next chirp its datagram 1420, even if the cell controller does not have enough time to identify the identity of the tag from the first received chirp of the datagram 1420. The cell controller can then anticipate this next chirp and ascertain the identity of the tag at that time. Once the tag is identified, the cell controller can duplicate the tag's pseudorandom number generator to calculate the times of all future chirps by the tag. In the tag datagram 1420 of Figure 12c, a series of special synchronization bits 1424 are inserted between the UID 1423 and the delay information 1425, to reliably determine when the UID 1423 ends; in this case, the UID 1423 must be defined so that it does not include the synchronization sequence or its inverse.

Figures 12a, 12b, and 12c include optional data sections 1404, 1415, 1426, which allow a tag to transmit data to the cell controller. These sections 1404, 1415, 1426 can include data from within the tag, such as from a motion detector or a low power indicator, or from an external device attached to the tag, such as metabolic information for medical telemetry if the tag is attached to a person.

An identifier preamble, related to the tag UID, precedes the tag UID. This identifier preamble enables the cell controller to quickly verify that a tag is chirping as expected, without needing to decode the tag's complete UID. This frees the cell controller for other activities, such as communicating with different tags in proximity to other antennas. The identifier preamble 1402, 1412, 1422 and tag UIDs 1403, 1413, 1423 are set externally, and if appropriate can be defined to include error correction bits.

The UID of a tag may be hardcoded into the tag (e.g., as a serial number). Tags may be grouped based on their UIDs, and different groups may be associated with different cell controllers. Each cell controller contains information (received from another source) about which tags are in the group associated with the cell controller. When a cell controller receives a tag signal, the cell controller can extract the UID information from the tag signal to determine whether the tag signal was sent by a tag in the group associated with the cell controller.

In the tag datagrams 1400, 1410, 1420 of Figures 12a-c, the delay information fields 1414, 1425 and data fields 1404, 1415, 1426 can also include error correction bits. For simplified processing, data can be reduced to a stream of half-bytes. To determine what value to send for a particular half-byte, the tag can look up the half-byte's value in a table which contains, for example, 8-bit values, which represent the value of the half-byte plus error correction information. A single cell controller can handle all three types of datagrams

1400, 1410, 1420 shown in Figures 12a-c. The choice of datagram type would depend on the application requirements for a particular tag.

The amount of time it takes for a cell controller to detect the presence of a tag may vary depending on the nature of the cell controller design. For example, a 100-microsecond time to switch antennas may be significant when the cell controller is cycling among 16 antennas. In order to be assured that a tag will be identified the first time its tag signal is received by the cell controller, the tag datagram header must be long enough to give the cell controller time to try all of its antennas. If the performance requirement is in the range of 100 tags per second, 2 or 3 extra milliseconds in the header can be tolerated. But for higher performance requirements, or when tag power consumption must be minimized, it is necessary to either improve the performance of the cell controller or to use a tag datagram 1420 of the type shown in Figure 12c.

By anticipating time of transmission from a particular tag, the cell controller can collect tag information from a variety of antennas in an organized way, in order to better calculate tag location by using antenna and/or frequency diversity. If a tag is responding exactly when it is expected to respond, it is not necessary for a cell controller to detect every bit transmitted in the tag datagram in order to be reasonably certain that it is receiving a signal from the correct tag. A correct identifier preamble arriving exactly on schedule is almost certain to be from the expected tag. This provides an opportunity for the cell controller to try a variety of antennas that may or may not be able to communicate with the tag.

If it is necessary to track tags between the tag datagram transmissions, the tag can be configured to send shorter transmissions more frequently. For example, if a tag is configured to chirp its datagram on average every 10 seconds, it may also be configured to transmit a much shorter code more frequently, such as every half-second. This shorter code

might be as short as one bit long, and take just a few microseconds to transmit. Thus, even hundreds of such transmissions per second would consume only a small percentage of the communication channel. The cell controller can anticipate the exact timing of each such transmission, thus matching each signal with the originating tag based on the time of the transmission. Error correction codes can be arranged such that long chirps from one tag will not typically be corrupted by quick chirps from other tags. The cell controller has the data to forecast most of such collisions.

When a tag is first introduced into a cell controller's range, collisions between datagram transmissions of different tags will occur in an unpredictable way. Transmission by a tag newly introduced into the area, or by a tag which spontaneously increases its transmission rate (such as in response to a motion detector or "panic button"), cannot be forecast by the cell controller 102a, and might cause data corruption. However, once the tag is identified, previous collisions can be modeled and questionable data discarded. Alternatively, if signals from two tags collide, the cell controller can select an antenna such that the signal the antenna receives from one tag will be stronger than the signal the antenna receives from the other tag.

In a more advanced tag design, the cell controller has a means for sending information and instructions to the tag during the times that the tag is known by the cell controller to be in operation. Such instructions can include commands to be passed on to a device attached to the tag. The cell controller is capable of downloading such information, most simply by on-off keying, or for more advanced tags by flipping the pseudonoise bit sequences to indicate one or zero. In general, the downlink (downloading) approach is driven by tag cost and feature requirements, with higher bit downlink transmission rates requiring more expensive receivers that consume more power. Thus, a single cell controller can support read-only tags, read/write tags, and high-speed read/write tags simultaneously,

with the cell controller adapting its behavior depending on the features supported in a particular tag.

Timing of transmission from tag to cell controller depends on the item being tagged. Inventory and equipment can be set to transmit relatively infrequently, such as once per minute. More frequent transmissions would be required, for example, for tags on people in a secure facility. For read/write versions of tags, timing of transmission could be modified on command from the cell controller.

Alternative tag designs enable variation in transmission time based on environmental factors. For example, motion detectors may be placed in a tag to decrease the time between transmissions when the tag is in motion. As another example, a tag might transmit more frequently and with higher power when the tag has been tampered with. As another example, a tag might incorporate a slightly modified Electronic Article Surveillance (EAS) device, which would cause the tag to transmit its UID more frequently when in range of a standard EAS detector. More generally, if a tag is attached to another electronic device, transmission interval can be modified under the control of that device.

Tag Power

Tags 101a-c transmit a low RF power level in order to increase their portability and lifespan. In addition, tag signal transmissions 107 are designed to be only a few milliseconds in duration. Therefore, even if a tag transponds its UID every few seconds, careful tag design allows the battery life of the tag to approximate the shelf life of the battery itself. For even lower power usage, a motion detector can be incorporated into the tag so that, for example, transmissions can be less frequent when the tag is stationary.

For some situations, battery replacement may be accomplished by incorporating the battery in an attachment mechanism. For example, re-usable tag electronics may be

attached to a disposable patient bracelet, with the battery included in the bracelet. As another example, a battery may be incorporated into the clip of an ID bracelet. More generally, a battery may be incorporated into an inexpensive disposable portion of an active RFID tag, with the electronics in the other, more expensive, portion.

If the item to which a tag is attached is a power source itself, the tag could tap into that power source. This approach is most practical in situations where the tag can he designed into the equipment itself (such as a handheld computer), or where the equipment and power source are large (such as a forklift). A larger power source allows for longer tag range.

Estimating Tag Location

A tag signal 107 is received at a time that is the sum of (1) known fixed delays in the cell controller 102a that transmitted the interrogator signal 106, due to its circuitry and the wiring to and from its antenna modules 104a-d, (2) fixed time delays in the antenna module 104a and tag 101a, and (3) the time for the interrogator signal 106 and tag signal 107 to travel through the air.

Since (1) and (2) are fixed, attention can now be turned to (3), the travel time of the interrogator signal 106 and tag signal 107 through the air. The duration of the pseudonoise sequence modulated onto the 244OMHz carrier signal 106 by the cell controller 102a must be greater than the combined travel time of the signal 106 and the tag signal 107. Techniques for correlating a pseudonoise sequence are well known in the art. In the absence of multipath effects, the cell controller 102a can derive a simple triangular correlation peak from the received tag signal 107, as shown in Figure 5B. But in most indoor environments, the actual received tag signal looks more like that shown in Figures 5D-5G. Indoor radio signals are subject to substantial multipath effects, due to reflection

from a variety of surfaces, such as whiteboards, fluorescent lights, file cabinets, elevator shafts, steel beams, and the like. When a tag 101a transmits a tag signal 107, the sum of the direct tag signal 107 and reflected signals is received at the cell controller antenna module 104a. A variety of approaches can be employed in order to extract a correlation peak from such information, with the particular approach chosen depending on the available signal quality, processing power, and required performance.

A 40-megahertz chip rate results in a correlation peak with a rise time of 25 nanoseconds, corresponding to a rise time distance of about 25 feet. Since tag location is calculated using round-trip travel time, single-chip accuracy therefore allows tag distance to be calculated within about 12 feet, without any advanced signal processing.

Approximate location of a tag can be calculated by noting when correlated signal-to-noise ratio rises above a predefined level. Improved accuracy can be achieved by trying a small variety of carrier frequencies and choosing the one that rises the earliest; such frequency diversity is supported by the radio system shown in Figures 4-6. This approach is sensitive to the system signal-to-noise ratio.

An alternative approach finds the peak of the correlation function. For an improved result, the signal delay is measured by taking the negative of the cross-correlation function's second derivative and finding the location of its peak, as shown in Figure 5G.

For highest accuracy, the MUSIC algorithm, known to those in this field, can be used, for which accuracy in the 0.01 chip range has been reported. MUSIC requires frequency diversity, which is supported by the radio system herein disclosed and shown in Figure 6. The method is based on a decomposition of the eigenvector space of the pseudonoise correlation matrix of the delay profile data vector. Frequency diversity is required, where each distinct frequency provides the information to solve for an additional multipath component. For tags that are mostly stationary, necessary data can be collected and the

calculation completed as a background process. For inventory applications, motion detectors can be incorporated into the tag, which would then inform the cell controllers whenever their locations need to be recalculated.

In situations where adequate data collection time, antenna diversity, frequency diversity, or processing power are not available, a variety of heuristic techniques can be used to calculate tag location, even in the presence of severe multipath effects. A variety of techniques well-known in the art can be used to estimate bearing to each antenna, which can be employed as required.

In many circumstances, although precise accuracy in calculating tag location is not required, it may still be important to calculate tag location with reference to a floor or partition. Story-by-story discrimination in a building can be accomplished by mounting antennas in the ceiling that radiate downward (or antennas in the floor that radiate upward), as shown in Figure 2. Similarly, antennas mounted sideways on horizontal partitions can determine location relative to those partitions. Relatively narrow beamwidth antennas, which are less sensitive to multipath effects, can be directed at doorways and the like.

A cell controller antenna 104a can be mounted near a computer screen, with coverage corresponding to viewing angle of the screen. Software can then be arranged to automatically configure the operating system for the person in range, or to blank the screen for security purposes depending on who is or is not in range. Similar concepts can be used in conjunction with copying machines, microfilm readers, restricted equipment, and the like.

A single antenna module can include three separate antennas placed in a triangle. By comparing phase difference using the Inphase and Quadrature components of the returning signal, an indication of the tag's angle can be determined. In high frequency

embodiments, such as at 2.45 gigahertz, such antennas could be within inches of each other and be quite effective.

Heuristic techniques can be used to analyze the correlation profile to estimate the time at which the correlation began, that is, the time at which the correlation peak begins to be distinguishable from the "noise" baseline. Frequency diversity can provide a variety of samples, the best of which can be chosen. Improved estimates can be achieved by pattern matching the correlation peak with one of a vocabulary of well-studied typical correlation profiles. For calibration purposes, tags can be placed at known fixed locations, and tags passing near those locations will likely demonstrate similar correlation profiles. Such fixed tags can also be utilized to detect the presence of jammers (objects which emit, intentionally or unintentionally, interfering signals) and to provide a real-time testbed for trying a variety of antijamming techniques.

Antenna diversity is the most important tool for improving the accuracy of tag location calculation. If low accuracy is required, antennas can be placed so that only one antenna is in range of a given tag. In this case there is insufficient data for triangulation, and only enough information to detect tag presence and estimate tag distance from that antenna. Approximate bearing of a tag can be estimated from the signal strength of antennas designed for the purpose and well-known to those skilled in the art; with the caveat that such bearing tends to reflect the strongest signal received, which might include a substantial multipath component. Conversely, for areas requiring high accuracy, a diversity of relatively narrow beamwidth antennas can be installed, for example, at entrances, which together provide a clear picture of location.

Antenna diversity also provides system scaleability. For facilities or sections within facilities, that do not require calculation of tag location with high accuracy, substantially omnidirectional and/or ceiling mounted antennas can be installed relatively far from each

other, for a relatively low cost per square foot of coverage. For facilities, or sections within facilities, that require high positional accuracy, a diversity of closely-spaced and/or directional antennas can provide high accuracy at an increased cost.

Cell Controller Operation

The design of the overall system, and the fact that the intervals between tag signal transmissions are generated pseudo-randomly, provide an opportunity for secure operation. Tags dispersed at the entrance of a secure facility can be tracked through that facility and a special code can be emitted when the tag is tampered with. Although the tag's code can be determined by monitoring the tag's response, the tag's transmission interval varies according to an algorithm that can be arranged to be known only to the tag and the host, and impossible to directly determine without destroying the tag. The tag may for example, include an element, such as a physical element, for reprogramming its code and transmission interval. For example, a photo ID with an incorporated tag might be reprogrammed each time the person wearing the photo ID passes a security checkpoint, potentially in conjunction with biometrics technology.

For full coverage of a facility, multiple cell controllers can be installed, covering somewhat overlapping areas. Although each cell controller will be operating according to a search and data collection method, rapid movement between antennas, the pseudonoise code, the changing chipping rate, and so forth, will appear as random noise to another cell controller. In addition, codes with known cross-correlation characteristics, such as Gold Codes, can be allocated to various cell controllers by the host computer, particularly the codes used to search for tags. Alternatively, cell controllers can switch choice of pseudonoise codes on a randomized basis.

For tags on the border between two cell controllers, each cell controller reports the tag's distances from its antenna modules. The central host 105 assembles this data to calculate the tag's location.

A variety of pseudonoise codes are available for use by a cell controller. Thus, if one code seems to be receiving interference from other users of the spectrum, the cell controller can choose another code. The tag, being in essence a transponder, does not need to know the particular code being used. Likewise, center frequency can be adjusted somewhat if another user is causing difficulty.

Digital Signal Processing

Referring back to Figure 10, the signal processing hardware 1004 performs operations upon the output of the digitizer 636. The signal processing hardware 1004 generally functions to perform operations upon the signal from digitizer 636 such as, for example, the previously described correlation. Additionally, signal processing hardware 1004 may perform a filtering process to filter out the noise components of received data signals. Various filtering techniques are known to those skilled in the art, such as the use of an adaptive transversal filter. Following is a description of functions that may be performed by the signal processing hardware 1004 in a preferred embodiment of the article tracking system of depicted in Figure 1.

Referring now to Figure 15A, shown is a block diagram of a preferred embodiment of the signal processing hardware 1004. Generally, the signal processing hardware may include one or more hardware components that collectively perform digital signal processing of the received signals. Recall that the signal processing hardware 1004 was previously described in connection with Figure 10. The processing set forth below may be

implemented using any combination of conventional off-the-shelf hardware and/or software, utilizing DSP and/or FPGA hardware and/or semi-custom or custom ASICs, that may be configured by one of ordinary skill in the art using the description set forth herein.

Shown in Figure 15A is signal processing hardware 1004 which is connected to the microprocessor 1001 and the digitizer 636, as previously described in connection with Figure 10. Included in the signal processing hardware is a correlator unit 1800 connected to other DSP hardware components 1800. The number and type of components that are included in the other DSP hardware components 1802 vary with the type of processing done in each particular implementation. Each of the hardware components included in the signal processing hardware may be controlled by the microprocessor 1001, such as by using connections 1804a and 1804b. The connections required vary with each particular implementation and associated processing.

One embodiment of signal processing hardware 1004 uses only the correlator 1800. In this instance, data is directly output by the correlation unit, as indicated by the path 1804a. Generally, the other components 1802 and associated connections, such as 1804b, would not be used in this embodiment. As will be described in paragraphs that follow in connection with Figure 15B, only a portion of the method steps of Figure 15B are performed by the signal processing hardware in this embodiment. In an alternate embodiment of the signal processing hardware which will also be described in paragraphs that follow, all of the method steps of Figure 15B are performed as part of digital signal processing.

Referring to Figure 15B, shown are method steps of an embodiment for filtering received signals to enable determination of when a tag signal has been received. At step 2000, a number of samples of received signals are taken. The number of samples to be taken varies with each particular implementation. Generally, in accordance with well

accepted principles known to those skilled in the art, the sampling rate is typically twice the transmission rate. For example, in one embodiment 254 points or received signals are sampled where each of the 254 signals is provided every 12.5 nanoseconds corresponding to the 80 megahertz sampling rate in which the system has a chipping rate of 40 megahertz. In this system, the signal transmitter rate is 127 chips per bit, or rather, every 3.2 microseconds a bit of data is transmitted. Generally, "chipping rate" and "chip time" are described in "The Practical Engineer", IEEE Spectrum, Vol. 35, No. 9, September 1998.

At step 2002, a first range of the sample received signals is determined. The first range is a subset of the samples recorded, as in step 2000. Generally, certain factors which are dependent upon each particular implementation may be considered when determining the starting point and size of the first range of step 2002. One factor that may be considered is the anticipated arrival time of a signal returned by a tag. This relates to, for example, anticipated delays in the transmission circuitry and may be used in determining a starting point of the first range.

Another factor that may be considered relates to the tag transmission range and the distance at which tags may be expected to be located. This may affect both the starting point of the first range as well as the size of the first range. For example, if objects in one system are known to be located within a small range, then the earliest possible time which a signal may be received by an antenna of a cell controller is earlier than a time of a different tag in another system in which objects are known to be located farther away from the transmission source. This may affect how many of the received signals which occur earlier in the sampling may be disregarded.

Generally, the data collection event, the transmission of the signals, and the anticipated hardware delays and other timing delays may be calibrated in accordance with the starting point of the collection or sampling. In one particular embodiment, due to the

calibration of the transmitting and the data collection as well as the anticipated delays, approximately the 70th sample was determined to be the beginning of the first range because this was one of the earliest points at which a return signal may be expected in a particular embodiment. Also in this embodiment, the actual span or size of the first range is 26 which was determined in accordance with the anticipated range of distances of objects to be located.

At step 2004 auto-correlation is performed for the first range of samples producing a magnitude for each sample in the first range. The performing of auto-correlation upon received signals was previously described. For example, Figure 5B shows the results of performing auto-correlation upon the wave form 500 as shown in Figure 5A. However, the waveform 510 of Figure 5B does not include any "noise" in addition to the originally transmitted signal 500 as shown in Figure 5A. One preferred embodiment implements the auto-correlation portion using a field programmable gate array (FPGA). The ability of the FPGA to perform massively parallel operations, such as matrix operations, efficiently in hardware is one factor in considering using the FPGA to perform the auto-correlation function of step 2004. Generally, use of particular hardware, such as the FPGA, may reduce the real time calculations and computational costs associated with performing expensive matrix operations, as in the method steps of Figure 15B.

Generally, in the calculation of the autocorrelation function, 254 consecutive samples of received data ("sample waveform") are compared with an idealized version of the same data ("reference waveform"). If a coherent demodulation is available, then a real correlation may be performed. Otherwise, a complex correlation may be performed. Autocorrelation may generally be defined as the integral:

$$\Psi(t) = \int_{-\infty}^{\infty} f(t)\ f(t-r)dt$$

The above equation is a measure of the similarity between a signal and a phase shifted replica of itself. An autocorrelation function is a plot of autocorrelation over all phase shifts (t-r) of the signal, where $\Delta t$, the change in time, is in half-chirp intervals.

For a 127-chip sequence, sampled once every half chip, each correlation calculation takes 254 multiply-and-add operations, and calculating the entire autocorrelation function takes $254^2$ which is approximately 64,000 multiply and add operations. If a complex reference waveform is used, computational complexity is increased by a factor of 4. Even with very fast hardware or specialized signal processing hardware, this number of calculations may cause a "bottleneck"
due to the amount of time required to perform the calculations.

A correlator implemented in hardware can generally make a quick estimate of tag location by combining various techniques, some as described above and other which will be described in the paragraphs that follow.

Since the reference waveform generally includes 1's and -1's, 2's complement arithmetic may be used for the multiplication operation. With this simplification, one of ordinary skill in the art may implement the foregoing using a field programmable array (FPGA) and/or semi-custom or custom ASICs, enabling operations to take place in parallel and with generally high throughput.

It is generally not necessary to calculate each interval of the autocorrelation function. Some of the range may be ignored because the tag is low-powered and can only be detected at a limited distance. For example, if the tag's radio has a maximum range of 100 meters, there is no reason to perform the autocorrelation function with phase shifts corresponding to the distances in excess of 100 meters. Additionally, in searching for the leading edge, phase shifts of a full chip or more may be used to search for the signal, and then half-chip intervals may be used in the neighborhood corresponding to the time when

the first signal is detected. More generally, a subset of the 254 autocorrelation offsets may be used in the search for the peak or the rising edge of the autocorrelation function This is described in more detail in paragraphs below.

At step 2006, heuristics are used to select a sample point which approximates where in the first range of received signals is the received tag signal. Generally, step 2006 produces a rough estimate as to the timing of the returned signal. This selected sample point is used, in connection with step 2008, to further refine and limit the sampled data points considered in determining the returned signal. One heuristic or technique which may be used to approximate the location in the first range of the received tag signal is related to the strength or magnitude of the signals within the first range. By looking for the autocorrelation peak in the first subset, this corresponding signal may be used in approximating where the received tag signal may be located. Generally, this is based upon the premise that the strongest received signal corresponds to the direct path of the received tag signal.

The rising or leading edge detection technique is a second heuristic that may be used to approximate where in the first range of receive signals is the actual receive tag signal. Generally, the samples are observed until a large or significant change in slope is detected. The actual determination of what is "large" or "significant" is relative to each system and varies with each implementation. One technique used with the rising or leading edge detection may include using a normalized value from 0.0 to 1.0 where 1.0 corresponds to the signal with the maximum amplitude received. When two points are encountered in which the slope of the line formed between these two points is greater than, for example, 20 percent of the normalized value, then this change may be considered large enough to signal a significant change in slope. Generally, the rising edge detection technique is based upon the assumption that the first peak is the line-of-sight returned signal. Note that this is

different than the premise or assumption of the first technique which is based upon the assumption that the strongest returned signal is the returned tag signal.

Yet another heuristic is a threshold detection technique. A threshold value is determined, and the first sample point having a magnitude greater than or equal to this threshold is the selected sample point. The threshold value chosen varies with environmental and implementation. Running trials of the system is one suggested method for choosing a threshold value.

At step 2008, a second range of samples is determined which is a subset of the first range of samples using the approximated location as determined in step 2006. In determining the second range of step 2008, the precise starting and end point as well as the size or span of the range must be determined. Both the span of the second range as well as the precise starting and ending points of the second range may be related to or dependent upon the heuristic used to approximate the location of the received tag signal in step 2006. For example, if the peak or maximum amplitude of the received signal were used in determining the approximate location of the receive tag signal, one common technique would be to take a specified number of equal points to the right and to the left of this peak and use this to correspond to the span and beginning and end points of the second range. If a different technique were used, such as the leading or rising edge detection technique, a varying number of points before and after the rising edge may be used. One preferred embodiment, for example, may use the rising edge detection technique or the threshold detection technique. In this embodiment, a range is determined having a starting point which is three to the left of the rising edge, and eight points to the right of the rising edge.

The second factor to be considered is the actual size of the second range. It should be noted that the size of the second range in one particular embodiment is 12. The reasons and factors that may be considered when choosing the size of the second range will become

more apparent in light of following paragraphs describing the different operations which are performed upon the second range of data.

In step 2010, the recursive least squares (RLS) technique using the second range of samples is performed. Generally, the RLS technique is used in the design of adaptive transversal filters and is based upon the least-means square adaptation method, as generally known to those skilled in the art. Functionally, the RLS technique used in step 2010 is used to filter out the noise component of received signals. In this embodiment, the RLS technique is used to filter out the noise components of the second range of sampled receive signals. The precise steps and how the RLS algorithm works are disclosed in the paragraph that follows.

At step 2012, a vector of filtered samples corresponding to the second range of samples being filtered are produced. As previously described, this vector contains values which correspond to filtered received signals.

At step 2014, the approximate peak corresponding to the filtered received tag signal is determined using the values included in the vector. Detail is described in the paragraphs that follow regarding how the peak is approximated using the values included in the vector produced by step 2012.

In step 2016, the tag distance is determined using the time of the filtered received tag signal. The precise details of how to determine tag distance using the time of the filtered received tag signal were previously described based upon the difference between transmission time and receipt time of the tag signal.

One of the functions of the signal processing hardware 1004 is to filter the noise component out of a received signal. Before describing detailed steps of how the RLS technique is used in performing this filtering process, a general description of how the RLS

technique is used in a feedback control system to perform this filtering function is described.

Referring now to Figure 16, shown is an example of an embodiment of a block diagram of a feedback control system which filters out noise components of received signals. Generally, the function of the block diagrams of Figure 16 combined filter out the noise components of a received signal. In this particular embodiment, Figure 16 depicts a feedback control system which includes a transversal filter 2022 and an adaptive weight control mechanism 2024. Generally, the transversal filter 2022 operates upon an input signal *u* and represented as an input vector u(n) with a varying number of components. The transversal filter produces an output signal E(n) which provides an estimate of the desired response d(n) 2028. In this instance, the desired response or signal d(n) is the actual received signal. The estimated signal E(n) produced by the filter is compared with the desired response signal 2028 to produce an error estimation $\alpha(n)$ 2030. This estimation error $\alpha(n)$ is the difference between the desired response signal d(n) and the estimated signal E(n) 2026 as produced as an output by the transversal filter. This error value $\alpha(n)$ 2030 is used as an input and feeds back into the adaptive weight control mechanism 2024. The adaptive weight control mechanism 2024 is a mechanism for performing the adaptive control process by varying certain parameters which feed back into the transversal filter 2022.

Generally, the transversal filter 2022 may also be referred to as a tap delay line filter. Further description and details of the transversal filter is described in paragraphs that follow in connection with Figure 17. In this particular embodiment, the RLS algorithm extends the use of the method of least squares to provide a recursive algorithm for the design of adaptive transversal filter such that, given the least squares estimate of the tap weight vector, w(n-1) of the filter at time n-1, the updated estimate of this vector at time n may be

computed. The estimated signal E(n) 2026 is denoted as $w^H$ (n-1) u(n). The precise notation of this will be described also in paragraphs that follow.

Referring now to Figure 17, shown is an example of one preferred embodiment of a transversal filter 2022. Generally, a transversal filter includes three basic elements: unit delay elements (2031a-2031m), multipliers (2032a-2032m-1), and adders (2033a-2033m-1). In particular, when unit delay operator 2031a operates on input u(n), the resulting output is $u(n)_1$. The role of the multiplier 2032a in the filter is to multiply the tap input 2030a by a filter coefficient referred to as a tap weight denoted $w^*_0$ (n). It should be noted that the asterisk in Figure 17 denotes complex conjugation which assumes that the tap inputs, and therefore the tap weights, are all complex values. The combined function of the adders in the filter is to sum the individual multiplier outputs and produce an overall filter component denoted y(i). In this particular embodiment shown in Figure 17, the number of delay elements is shown as m-1. This is commonly referred to as the order of the filter. Each of the components 2031b-2031m-1 operates in a manner similar to 2031a. Similarly, multipliers 2032b-2032m-1 operate similarly to the multiplier 2032a. Generally, the input signal u(n), and the tap weights denoted w(n) are represented as vectors with each element of the vectors corresponding to various components. When the various components are summed, they produce an estimated signal denoted y(i) in Figure 17. u(n) denotes the tap input vector at a particular time n. The tap weight vector w(n) defines the tap weight vector at a particular time n.

Generally, the recursive least squares or RLS technique attempts to choose a tap vector w(n) which minimizes the expected squared error. The error is determined as the sum of the differences between the expected or actual signal and the estimated signal output from the filter. The adaptive weight control mechanism 2024 of Figure 17 is used to determine the weighting factor associated with each of the errors at a particular point in time

where the error is the difference between the desired response or the actual output signal, and the estimated signal produced by the filter. The use of the weighting factor is intended to ensure that data in the distant past is "forgotten" in order to afford the possibility of following the statistical variations of the observable data when the filter operates in a non-stationary environment. Thus, use of the weighting factor allows additional weight to be given to the error values which are most recent in time, and give less weight to those error values which are earliest in time. The precise use of a weighting factor and how it relates to the measure of the "memory" of the RLS technique will become more apparent in following text.

Generally, the adaptive weight control mechanism 2024 provides a correction factor which is applied to the tap weights upon subsequent processing of data. In other words, a correction factor determined at time n is applied to the tap weight at a time of n+1. As known to those skilled in the art, a scaled version of the inner product of the estimation error and the tap input denoted u(n-k) is computed for k=0,1,2... to m-1. The result obtained defines this correction factor which is applied to the tap weights. Thus, the error factor provides for adjustment or correction of the various tap weights. This is the nature of the feed back mechanism of the system 2018 of Figure 16.

The previous descriptions regarding Figures 16 and 17 present a general description of the transversal filter and its corresponding adaptive control mechanism as generally known to those skilled in the art of adaptive filter theories as may be used in digital signal processing, for example.

Now what will be described is the RLS algorithm as generally set forth in the textbook entitled "Adaptive Filter Theory", by Simon Haykin, ©1986 by Prentice Hall, Inc.

Referring now to Figure 18, shown are method steps of an embodiment for performing the recursive least squares (RLS) method. At step 2034, variables are initialized

to initial conditions. Generally, this includes variables used in subsequent method steps such as the loop control variable n. Some of the variables used in Figure 18 processing steps are initialized to a set of initial conditions. Specifically, n is initialized to one in the method described in Figure 18. The tap weight vector w and the tap input vector u may be initialized to 0. The inverse of the correlation matrix denoted P(0) may be initialized to the inverse of a small positive constant time the identity matrix denoted I. The recommended choice of the small positive constant $\lambda$ is that it should be small when compared to 0.01 times the variance of the data sample u(n).

Generally, $\lambda$ is a small positive constant which may be referred to as the "forgetting factor". Since this constant is between zero and unity, multiplying a variable by $\lambda$ reduces the magnitude of the variable. Using the RLS technique, a weighted sum of square errors is minimized by choosing a vector of coefficients. In this summation, the squared errors in the "past" are weighted by higher powers of $\lambda$ that are more "recent" squared errors. Thus, the coefficients produced by the RLS technique is generally chosen with less regard for errors in the past.

There are many "rules of thumb" for choosing $\lambda$ which may depend upon the variance of the data, the variance of the additive noise, the number of components to be determined, and the rate of change of the underlying system to be estimated. It may generally be helpful to note that the effective number of errors which are not yet forgotten is approximately $1/(1-\lambda)$. A forgetting factor of 1/100, for example, yields roughly 100 significant errors. If the underlying system to be estimated is time invariant, then $\lambda$ should be set as close as possible to unity. In the instance of a time-varying system, $\lambda$ should generally be chosen such that:

$$\frac{1}{1-\lambda} > 3 * (\textit{the number of coefficients estimated})$$

and such that generally:

$$\frac{1}{1-\lambda} \leq \textit{the number of samples for which system is approximately constant.}$$

The former condition generally reflects the fact that the RLS technique provides a converged solution in 3 to 5 times the number of coefficients for cases in which the additive noise is not too severe. The latter condition reflects the fact that the RLS technique attempts to find, for each sample input, a single vector of coefficients which may be used to approximate the last 1/1-$\lambda$

outputs of the time-varying system.

In accordance with principles described in the Haykin textbook, certain factors should be considered when initializing values for use with the RLS method. In particular, regarding a starting value for P(0), a starting value should be chosen which assures the non-singularity of the correlation matrix.

After initialization, control proceeds to step 236 where a determination is made whether or not the loop control variable n is less than or equal to the number of desired iterations. If the determination is made that n is greater than the number of desired iterations, meaning that execution of the method steps of Figure 18 is complete, control proceeds to step 238 where the method depicted in Figure 18 stops. If a determination is made at step 2036 that n is less than or equal to the number of iterations, then control proceeds to step 2040. At step 2040, the gain denoted k(n) is computed as:

$$k(n) = \frac{\lambda^{-1} \; P(n-1) \; u(n)}{1 + \lambda^{-1} \; u^{H}(n) \; P(n-1) \; u(n)}$$

Control proceeds to step 2042 where the a priori estimation error denoted $\alpha(n)$ is computed. $\alpha(n)$ is computed as:

$$\alpha(n) = d(n) - w^{H}(n-1)\ u(n)$$

Control proceeds to step 2044 where the tap weight vector for a particular instance in time denoted w(n) is computed as:

$$w(n) = w(n-1) + k(n)\alpha^{*}(n)$$

Control proceeds to step 2046 where the inverse of the correlation matrix denoted P(n) is computed as:

$$P(n) = \lambda^{-1}\ P(n-1)\ -\lambda^{-1}$$

Control then proceeds to step 2048 where the loop control variable n is incremented by 1. Control proceeds to the top of the loop at step 2036 where a determination again is made whether or not the loop formed by steps 236-248 has been performed the desired number of iterations.

It should be noted that the method step 2044 describes the adaptive operation of this method whereby the tap weight vector w is updated by incrementing its old value by an amount equal to the complex conjugate (denoted by an *) of the a priori estimation error $\alpha(n)$ times the time varying gain vector k(n), hence the name "gain vector". The a priori estimation, denoted $\alpha(n)$ represents the a priori estimation error. Generally the a priori estimation error refers to an estimate of the error based on a tap weight vector that was made at time n-1. The constant $\lambda$ is a value close to 1. $1-\lambda$ "represents a measure of the memory of the algorithm". As previously described, this is a weighted value introduced in

a definition of the cost function based on the error at time n. Generally, $\lambda$ is a positive constant close to but less than 1. When $\lambda$ equals 1, we have the ordinary method of least squares. The inverse of 1 minus $\lambda$, generally speaking, is a measure of the memory of the algorithm. The special case where $\lambda$ equals 1 corresponds to having infinite memory. As previously described in accordance with the use of a weighting factor being associated with the error at each particular point in time, $\lambda$ is used in determining the actual weight given to particular values of the error determined at different points in time.

It should generally be noted that this algorithm is deemed to be "recursive" for the fact that updating the tap weight vector at time n, a prior value for the tap weight vector at time n-1 is used. This becomes apparent when method step 2044 is examined where the tap weight vector denoted w(n) is computed as being dependent upon the value w(n-1). It should also be noted that in the flowchart of Figure 18, the asterisk denotes the complex conjugate of a number. Additionally, the superscript of H, as depicted in step 2042 when associated with the tap weight vector w, implies that the tap weight vector w has the Hermitian property. Generally, a complex valued matrix such as the tap weight vector w(n) is Hermitian if it is equal to its conjugate transpose, as known to those skilled in the art.

Referring back to Figure 15, step 2010 performs the RLS technique just described using the second range of sample receive signals. Also recall that a note was made that the size or span of the second range may generally be related to use of the RLS technique. The RLS technique performs matrix operations which are generally expensive in terms of computing time and resources. Thus, this expense is often a factor to be considered when determining the size of the second range in that the size of the second range affects the dimensions of the matrix and hence the number of matrix operations which are performed in computations of the RLS technique for the second range of samples. Additional factors should also be considered when choosing the size of the second range. Generally, if the

number of points considered in the second range is too large, and the noise component of the receive signal includes tonal frequencies or jammers, then the model may follow the peaks of the jammer signals rather than properly fit a curve identifying the filtered received signal. Additionally, if the size of the second range is too small, then enough points may not be considered to properly fit the curve. It should generally be noted that the number of points or data samples to be included in the second range varies with the system and implementation.

The RLS technique used in this embodiment assumes that the observed sequence includes a linear combination of a known number of the data sequence. Generally, the RLS technique attempts to find the combining coefficients for this linear combination which best fits the observation. Generally, the RLS technique assumes a wide-sense stationary process. Generally, the RLS technique also presumes the presence of random additive noise which is uncorrelated with the data sequence. With the presence of "white noise", the sum process remains generally wide-sense stationary. However, with the addition of a tonal frequency or jammer frequency, the sum process is no longer wide-sense stationary. To use the RLS technique when the sum process is not wide-sense stationary, such as may be in the application of this embodiment as used indoors, a corrective factor should generally be considered. One corrective technique that may be considered is in choosing the number of coefficients or taps.

In this instance, with the sum process not being wide-sense stationary, care should generally be taken to insure that the number of coefficients are not overspecified when using the RLS technique. In this embodiment, the noise input to the correlator may be modeled as the superimposition of two components: a wide-sense stationary white noise process, and a non-stationary intermittent plurality of tones of unknown frequencies and amplitudes. Based on this model for this particular embodiment, it is found that the number of

coefficients or taps should generally span the main peak of the impulse response magnitude. However, the number of taps should generally not span more than this in this particular embodiment.

Generally, the number of taps may vary with embodiment and application. For each application and embodiment, the observed process should be modeled to take into account all factors and a number of taps chosen in accordance with these considerations as described herein.

As just described, factors that may be considered when choosing a size for the second range include consideration of the time complexity regarding computational expense as well as application for real time considerations when performing complex calculations. Additionally, the size of the second range varies with the environment in which this application will be used. If there will be tonal frequencies, such as a microwave oven within an indoor environment, this should be considered when choosing an appropriate value for the second range. In one preferred embodiment, a value of 12 was used for the size of the second range for an indoor article tracking system which included the previously described transmission rate of 127 bits per second in a sample of 254 points of received. This indoor system may possibly have tonal frequencies and other jammer signals since it is an indoor application as well as multipath noise. Thus, a range size of 12 for the second range was used considering these factors.

Generally, the steps describing the RLS technique of Figure 18 may be performed using a different data sample set for each execution or iteration of the loop. In one particular embodiment, one data set of 254 received signals was recorded. Rather than perform the method steps of Figure 18 with a different set of 254 data samples each time the method steps of Figure 18 were performed for a particular iteration, the same data set is used for each iteration. Ideally, the number of iterations of the RLS algorithm should be as

large as possible in order to meet the mathematical convergence for simulating when n goes to infinity. However, in one preferred embodiment, the method steps of Figure 18, as described in the loop formed by steps 236 through 248, are performed for 84 iterations. Through experimentation for this particular embodiment, it was determined that this was an optimal value to be used in the tradeoff of real time application, computational complexity, and accuracy of locating an object. For other systems and other applications, this number may vary.

As an output of the RLS method performed in step 2010, a vector of filtered samples is produced in step 2012. Each element in this vector corresponds to a component of a received signal which represents a filtered signal. In other words, each element of the vector corresponds to a filtered signal with the noise portion removed. The vector produced in step 2012 is a vector of filtered signals in which each element of the vector corresponds to an element of the second range. The RLS technique performed in step 2010 removes the noise component and returns a filltered signal.

At step 2014, using the data points included in the vector of step 2012, a technique is applied which approximates the peak corresponding to the filtered received tag signal. A more detailed description of step 2014 is set forth in paragraphs that follow in connection with Figure 20.

Referring now to Figures 19A-19E, shown are sample waveforms of received signals for a number of sample points. In Figure 19A, shown is an example of a correlated received signal with a small amount of "white noise". It should be noted that the waveform of Figure 19A generally does not include a multipath component, as would be seen in the system of the embodiment described herein. Figure 19B shows a waveform with a low degree of indoor multipath components as may be included in a received signal. Figure 19C illustrates a waveform that includes a medium degree of severity of indoor multipath noise

in addition to a transmitted signal, and Figure 19D shows a severe amount of indoor multipath noise added to a transmitted signal. Each of the waveforms shown in Figures 19B-D are waveforms that may be received in different environments within which a preferred embodiment of the invention operates. Shown in Figure 19E is a correlated waveform, labeled "correlation", which is also typical as the output waveform after performing step 2004 of Figure 15. The second waveform, labeled "model" in Figure 19E, is an example of a waveform resulting from graphing the 12 points included in the second range of samples which is output from the RLS technique in one preferred embodiment. As previously described, this vector may be produced in step 2012 after performing the RLS technique in step 2010. The peak of the actual waveform, for example, may be approximated in step 2014, as the 4$^{th}$ data point of the "model" waveform of Figure 19E.

In this particular embodiment, the transmission rate is 127 chips per data bit. In other words, the bit pattern of the transmitted signal repeats itself every 127 bits. A lesser number bit sequence such as 31 chips, may also be used in a particular implementation which requires less processing time and less number of samples to be taken. However, the gain is lost as you decrease the number of patterns in the bit sequence. A longer bit sequence generally allows for a greater distribution of energy over a larger period of time. A longer sequence generally enables a stronger signal at the output of the correlator than that produced by a shorter sequence. The processing gain indicates this signal strength enhancement and refers to the number of bits in the sequence.

Other factors regarding a bit sequence which should be considered when implementing the techniques described herein relate to the properties of the bit sequence regarding auto-correlation and cross-correlation. For example, a maximal sequence has good auto-correlation but bad cross-correlation. Thus, a transmitted signal which is a maximal sequence would produce an idealized peak waveform, for example as shown in

Figure 5A. However, the cross-correlation of such a sequence may not be desirable in an embodiment where better cross-correlation is required. For example, in an embodiment where one is required to detect or reduce the interference between multiple transmitted signals, good cross-correlation is often needed. For applications such as this where one is required to distinguish between multiply transmitted signals, a sequence such as the Gold code may also be used. It should be noted that the precise sequence length as well as the various properties of the sequence sent in the transmission signals may vary with application and each particular implementation.

In the previously described embodiment, a local positioning system was described in which the assumption is made that the articles or persons being tracked by the system may be at a different position at any particular point in time. In other words, there is no assumption that the person or object will remain stationary for a majority of time. In an application of a system which includes the local positioning system and the techniques described herein, if the objects being located are primarily stationary for a majority of the time, then special processing may be performed in the atypical case when an object is determined or sensed to have moved, as by a motion detector. When motion of a primarily stationary device has been detected, special processing may be performed using the techniques previously described. Since it is the atypical case in which the location of an object is to be determined, in the small number of instances when the location of an object or person needs to be determined, the system may devote additional processing time to locating the object. Thus, in an application of article tracking in which the objects tend to remain stationary, minimizing the amount of computational time to the extent as previously described may not be a factor in selecting, for example, the size or span of the second range.

It should be noted that various portions of the signal processing hardware 1004 may be implemented in varying combinations of software and/or hardware dependent upon the

particular application and system implemented. For example, in one preferred embodiment, the auto-correlation function performed in step 2004 on a first range of samples is implemented in a FPGA, as previously described. Dependent upon each particular system, application, and requirements of each system, other functions of the signal processing hardware 1004 as described herein may be implemented in varying combinations of software and/or hardware.

In one of the previously described embodiments, the same data set was used when performing multiple iterations of the RLS algorithm of Figure 18. As an alternative to using the same data set for each iteration, a single data set set may be "reused" on subsequent iterations with slight modifications, such as rotating or shifting to the right or to the left by one data element for each iteration.

Referring now to Figure 20, shown are example steps of a method for approximating the peak corresponding to the filtered receive tag signal using the values included in the output signal vector produced as a result of step 2014. The method steps shown in Figure 20 are more detailed steps of the method step previously described for step 2014 of Figure 15. For all of the magnitudes of the signals included in the output signal vector, determine the largest magnitude, as in step 2100. It should be noted that in one embodiment, if the first or last element of the vector has the largest magnitude, then an assumption is made that the received signal cannot reliably be determined, and the method stops execution.

Subsequently, in step 2102, a threshold value is determined which, in one embodiment is equal to 62.5 percent of this largest magnitude previously determined in step 2100. At step 2104, it is then determined which element in the output signal vector is first in time to exceed the threshold value . This signal determined to be first in time to exceed the established threshold is referred to as the X. It should be noted that in one preferred

embodiment, if the "X+1$^{th}$" element of the vector is smaller than the magnitude of the "X$^{th}$" element, then the "X-1$^{st}$" element is used rather than the "Xth" element.

At step 2106, other elements of the vector are selected to be used in subsequent processing steps to determine the actual received signal. In one embodiment, two other elements are generally chosen. These are the vector elements denoted by the indices "X+1" and "X+2". It should be noted that the element indicated by the index value "X+2" is not used if the the "Xth" element is the second to last vector element. In step 2108, using a weighted average formula of these 3 points, the actual received tag signal is determined based on the expected shape of the received signal as being a correlated signal with a triangular peak. One embodiment calculates or estimates the received tag signal as indicated in the pseudo-code type description below. It should be noted that the description below generally summarizes that which is set forth and previously described in conjunction with the method steps of Figure 20.

```
Total = 0
FOR each element in the vector, v, DO
{
Determine the weight of current vector element, denoted by index=j, as:
        weight v[j] = MAX(0,  magnitude(vector element j) -
                              (largest magnitude of all vector elements/4) )
Total = total + weight v[j]
}
Received signal = X+1 - (weightv[X]/Total) + (weight v[X+2]/Total)
    or
Received signal = [X+1 - (weightv[X]/Total)  /** if element X+2 is not
                                                 used and is the  second to last
                                                 element of the vector **/
```

The following points are worth noting regarding the previously described embodiments of the signal processing hardware 1004 and previously described method

steps of Figure 15B. In one of the previously described embodiments of the signal processing hardware which includes only the correlator 1800, method steps 2008-2014 of Figure 15B are not performed. Rather, only method steps 2000-2006, and 2016 of Figure 15B are performed. Specifically, in this embodiment the autocorrelation (step 2004) is performed by the correlator 1800. The approximation produced as a result of using the heuristics at step 2006 is considered to be the received tag signal. A first heurisitic, as used in step 2006, determines the received tag signal to be the signal in the first range with the maximum magnitude of all the signals in the first range. A second or alternate heurisitic that may be included in an embodiment (at step 2006) is to choose a threshold value. The first signal included in the first range having a magnitude equal to or exceeding this magnitude is determined to be the received tag signal. The method for choosing the threshold value may vary with environmental and other factors particular to each implementation. Generally, this threshold value is selected in accordance with trial test runs of particular implementations to allow for "tuning" the threshold value. Using either heuristic, this received tag signal is then used (step 2016) in determining the tag distance.

Other embodiments of the signal processing hardware 1004 may include other hardware components in accordance with the particular digital signal processing requirements in a particular embodiment. Each particular embodiment may be implemented in using a variety of combinations of hardware components, including, but not limited to, gate arrays and read-only-memory. Additionally, other embodiments of the signal processing hardware may be implemented as some combination of hardware and software in which the machine executable code may be executed on a computer system, such as the microprocessor 1001 or the host computer 105 or yet another computer component included in the signal processing hardware 1004 as a dedicated processing unit. The components may vary with application and design choices associated with a particular implementation.

Another embodiment of the signal processing hardware includes hardware and/or software in addition to the correlator. This embodiment may perform, for example, all the method steps of Figure 15B, rather than only some of the steps of Figure 15B, as in the embodiment with signal processing hardware that includes only the correlator.

The foregoing description sets forth a technique using an RLS method that affords a flexible and efficient way of filtering noise from a received signal as used in an article tracking system. Both multipath noise and tonal frequencies may be taken into consideration as factors when using the foregoing techniques in various environments within which an article tracking system as described herein may operate.

The foregoing technique for filtering a received signal is scalable for use in applications with both large and small sample sets in a variety of different environments each having different "noise" considerations while generally providing a high degree of accuracy in locating objects as required in article tracking systems as described herein.

Other embodiments are within the scope of the following claims. For example, the order of performing steps of the invention may be changed by those practiced in the field and still achieve desirable results.

An alternative embodiment includes modifications to the previously described cell controller and tag that are described in paragraphs that follow.

Chassis General RF description

The chassis RF subsystem consists of three of the five major cell controller components: the cell controller transmitter module, cell controller receive module and the single-pole, 4 throw switch modules. The digital section and the controller section are described elsewhere. This description outlines the operation of all three elements and presents performance data.

The function of the cell controller transmitter is to modulate the digital baseband spreading signals onto the transmitter carrier to generate the direct-sequence spread-spectrum RF signal. This transmitted signal must comply with the FCC Part 15 regulations, specifically with sections 15.247, 15.209 and 15.205. The transmitted RF signal power level must be digitally adjustable over a range commensurate with the variation in path loss of the transmitted signal to the tag (about 32dB at 2.44GHz over 125 feet). The final radiated signal should have an effective isotropic radiated power level of about 30dBm (1 watt).

The function of the cell controller receiver is to convert the received signal delivered by the antenna to an intermediate frequency, demodulate the spread-spectrum signal from the tag and deliver a filtered baseband signal to the analog to digital converter. The receiver also implements automatic gain control over a 50dB range at the intermediate frequency. The function of the single-pole, four throw switch module is to properly route the combined transmit / receive / DC signals to the correct antenna.

The cell controller RF elements have some common features that are digitally controlled by signals from the microprocessor. The transmit and receive local oscillators (which control upconversion and downconversion carriers) are set digitally. The transmit power control is set with digital signals so that the total received signal SNR can be maximized over the entire path traverse of a tag (from right on top of a particular cell controller antenna to maximum distance from that antenna). The receiver gain level is digitally measured and delivered to the baseband signal processor. Finally, the transmit and receive signals are combined onto one wire (which includes DC power) and this combined signal is switched from antenna to antenna by digital control from the microprocessor. Up to sixteen antennas may be installed on each cell controller.

Cell Controller RF Description

General Description

Referring now to Figure 21, shown is a block diagram of an embodiment of a cell controller. The cell controller RF elements can best be understood by referring first to the block diagram of Figure 21. The first item to note is the cell controller RF elements include active components in the antenna. This is because the signals sent up to the antenna are low in level and widely spaced in frequency. The RF cable transports the low-level Tx signal at 2442MHz to the antenna as well as the low-level Rx signal at 5800MHz from the antenna to the receive module in the cell controller. The cable also supplies DC power to the antenna from the chassis. Note that the transmitter module provides the first stage of RF switching to the antennas: it incorporates the "bank" switches that select which of the four SP4T (Single Pole, Four Throw) switch modules are to be used. Additionally, it incorporates the diplexer that separates transmitted signals from received signals and also a stage of Rx signal amplification.

The cell controller receive module will be briefly presented to show the frequency plan and signal flow. Also, the operation of the active antenna will be presented in detail.

Transmitter Module

In one embodiment, the transmitter module may be included in a card that fits into the main backplane of a 3D-iD chassis and provides the necessary transmitter signals. It receives DC power from the backplane. It uses input signals from the digital module and provides signals to the receive module and the digital module. It provides transmitted signals to the active antennas through SP4T switch modules installed in the chassis.

Modulator 2200

The digital baseband signal that represents spread data is an input to the transmit module. It modulates a 360MHz signal (provided by the 360MHz LO synthesizer module) in the biphase modulator consisting of a driver and a mixer. This modulator generates a classic Sin(x)/x spectrum at 360MHz and provides a signal level of about -7dBm to the next stage.

IF Local Oscillator 2204

The intermediate frequency (360MHz) local oscillator provides a carrier for the modulator. It consists of a commercial voltage-controlled oscillator, a buffer amplifier, a programmable synthesizer and a lowpass filter. The synthesizer is programmed by the microprocessor and returns its lock-detect status. The nominal output level for this LO is +6dBm.

SAW Bandpass Filter 2202

The function of the surface acoustic wave bandpass filter is to pass through only the main lobe of the modulated signals and to remove any sideband signals. This device is a passive filter and is used unmatched. This filter provides most of the suppression of out-of-band signals needed to comply with the Part 15 regulations.

Transmit Level Control Attenuator 2206

The function of the transmit level control is to set the final antenna radiated output power to the desired level. This can be done antenna by antenna and can be tailored to the installation site. The microcontroller allows the installer to adjust each antenna's output

power and stores these settings in a configuration table. As the cell controller operates each antenna in turn, it sets the transmit attenuator to the configuration value determined at installation. The transmit level control is implemented with a digitally controlled attenuator that has a 31dB range in 1 dB steps. The attenuator is followed with a buffer amplifier.

Transmit Mixer and Bandpass Filter 2208

The transmitter intermediate frequency signal is upconverted from the intermediate frequency to the final RF frequency in the transmit mixer. This mixer uses a local oscillator signal at 2082MHz to convert the 360MHz transmit IF signal to 2442MHz, the final RF frequency. The nominal LO level is +9dBm and the mixer suffers about 6dB of conversion loss. The mixer is followed by a 3-pole ceramic bandpass filter that serves to pass only the desired upper sideband and to reject the lower sideband and spurious signals.

RF Transmit Local Oscillator 2210

The RF local oscillator operates at 2082MHz and provides a carrier for the transmit mixer. It consists of a commercial voltage-controlled oscillator, a buffer amplifier, a programmable synthesizer and a power divider. The synthesizer is programmed by the microprocessor and returns its lock-detect status. The nominal output level for this LO is +9dBm. This synthesizer also provides the input tone for the receive module downconverter local oscillator (hence the power divider).

Transmit Preamplifier and Bandpass Filter 2212

The preamplifier and bandpass filter serve to boost the RF transmit signal to a

level sufficient to drive the final transmit amplifier without compromising signal linearity. The bandpass filter serves to help clean up mixer spurious and to help remove harmonics. The preamplifier consists of two stages that have a total gain of about 33dB. The amplifiers used in the preamp are identical to the buffer amplifier used in the transmit local oscillator. The bandpass filter is identical to the filter following the transmit mixer.

Transmit Final Amplifier 2214

The final transmit amplifier serves to boost the transmitted RF signal to prepare for the ride through the various switches and the RF cable on its way to the active antenna. It consists of a pair of amplifiers in a balanced amplifier configuration using hybrid couplers. The gain of this stage is about 17dB and this amplifier provides a signal of about +10dBm (at the maximum Tx level setting). The 1dB compression point of this amplifier is about +19dBm. The amplifier is implemented with devices identical to the Tx preamp and the LO buffer. The hybrid couplers are implemented with the same power divider used in the local oscillator.

Diplexer 2216

The function of the diplexer is to provide separate paths for the transmit and receive signals so they can be combined onto one coaxial cable. The diplexer is implemented with microstrip technology as a lowpass-high pass filter set with a common feedpoint. The isolation requirement of each branch of the diplexer is eased by the extensive use of other bandpass filters in the transmit, receive and antenna modules. This filter is fabricated on the printed circuit board.

Single Pole, Four Throw Bank Switch

The circuitry in this section 2230 serves to route the combined Rx / Tx signals to one of four switch banks. This switch is implemented as a cascade of three GaAs single-pole, double-throw RF switches that are controlled by digital signals from another cell controller module (digital module).

Single Pole, Four Throw Switch Module

The SP4T module is a separate plug-in unit that fits into a small backplane in the 3D-iD chassis. One module is required for the chassis, enabling 4 antennas to be connected 2232a-d to the system. Up to 4 SP4T modules may be installed in the chassis, providing service for up to 16 antennas. The switch module RF configuration is identical to the single-pole, four throw bank switch described above. Additionally, the SP4T module implements switching to impose DC on the output ports after the RF signal has been routed. Bias tee circuitry is used to isolate the DC switches from the RF circuitry (to prevent unnecessary RF loading). Note that the number of these modules varies in accordance with the number of antennae in a particular embodiment.

Receive Preamplifier 2220

The transmit module contains the first receive preamplifier following the diplexer. This architecture was chosen because the receive signal needs to leave the diplexer and go to the receiver module through a cable, so some signal stabilization was needed. This preamplifier is implemented as a balanced amplifier with GaAs units. It has a gain of about 19dB and the balanced configuration guarantees a good impedance to the diplexer. The hybrids for this amplifier are implemented as traces on the printed circuit board.

10IMHz Reference Generator 2222

Each of the phase-locked local oscillators used in the 3D-iD chassis requires a high-stability reference signal. This section provides a reference signal for the 360MHz IF LO, for the 2082MHz LO and for synthesizers in the receive module. This reference generator is implemented with a commercial, high-stability crystal oscillator at 10MHz followed by buffer amplifiers, a bandpass filter and a four-way power divider. The nominal output level on each of the four ports is 0dBm.

Antenna Module 2300

Referring now to Figure 22, shown is a block diagram of an embodiment of an antenna module. The antenna module 2300 includes a unit that mounts on the wall or the ceiling to provide the necessary radiated transmit and receive signals for a 3D-iD tag. In addition to the radiating antenna arrays, the module has amplifiers and filters mounted on a printed circuit board that also provides the ground plane for the antenna arrays. All inputs and outputs for this module arrive on a 0.25" coaxial RF cable. The printed circuit board is mounted on an aluminum backing plate and has a thermoplastic radome over the antenna arrays.

Antenna Bias Tee 2302

The bias tee extracts the DC current from the coaxial cable to power the active components on the antenna module. It provides a nominal 12 volts to the regulators on the antenna module that in turn provide 8 volts and 5 volts to the appropriate amplifiers.

Antenna Diplexer 2304

The diplexer is identical to the diplexer used in the transmit module.

Antenna Transmit Preamplifier 2306

The transmit preamplifier serves to boost the low-level transmitter signal delivered by the coaxial cable to a level sufficient to properly drive the antenna final transmit amplifier. The preamp is implemented as two amplifiers that are identical to the LO buffer amplifier in the transmit module.

Antenna Transmit Final Amplifier 2308

The antenna transmit final amplifier is implemented as a balanced amplifier using a topology and components identical to the transmit module final amplifier described above.

Antenna Transmit Final Bandpass Filter 2310

The antenna transmit final bandpass filter is a hairpin bandpass filter centered at 5800 MHz with a passband bandwidth of about 100MHz. The combination of the responses of this filter and the filters in the transmitter module ensure that the radiated transmitted RF signal conforms to the requirements of Part 15.247 and 15.209, especially at the allocated band edges.

Antenna Transmit Antenna Array 2312

The transmit radiating elements of the antenna module are a series-fed patch array (2 elements) whose center frequency is 2440MHz. The 3D-iD tag uses a low-cost, linearly polarized patch antenna. To prevent tag signal loss due to antenna cross-polarization for tags that are not aligned with the active antenna, the radiating elements here are circularly polarized. This choice of antenna polarization results in an additional 3dB of effective path

loss between the active antenna and the tag antenna. The gain of the transmit antenna array is about 11dBi.

Antenna Receive Antenna Array 2314

The receive radiating elements of the antenna module are a series-fed patch array (3 elements) whose center frequency is 5770MHz. The 3D-iD tag uses a low-cost, linearly polarized patch antenna. To prevent tag signal loss due to antenna cross-polarization for tags that are not aligned with the active antenna, the radiating elements here are circularly polarized. This choice of antenna polarization results in an additional 3dB of effective path loss between the active antenna and the tag antenna. The gain of the receive antenna array is about 12dBi.

Antenna Receive Bandpass Filter 2316

The antenna receive bandpass filter is a 3-pole ceramic bandpass filter centered at 5770MHz with a passband bandwidth of about 100MHz. This filter serves primarily to ensure that only the desired tag signals are brought into the receive chain.

Antenna Receive Low Noise Amplifier (LNA) 2318

The antenna receive low noise amplifier is identical to the receive preamplifier described above. It is implemented as a balanced amplifier to present a good impedance to the Rx bandpass filter. The hybrids for this amplifier are implemented as traces on the printed circuit board.

Antenna Receive Buffer Amplifier

The antenna receive buffer amplifier provides enough gain to compensate for the loss of the coaxial cable that ferries the received signal to the receiver module in the cell controller. It is implemented with two stages, each having an active device identical to the one used in the low-noise amplifier. The final stage has an impedance good enough to drive the diplexer without mismatch.

Receiver Module

The receiver module is a card that fits into the main backplane of a 3D-iD chassis. It receives DC power from the backplane. It uses input signals from the digital module and the transmit module. It provides signals demodulated baseband signals to the digital module for correlation.

Receive Final RF Amplifier

The receive final RF amplifier recovers the signal level lost over the RF coaxial cable between the antenna and the chassis. It is preceded by a 5800MHz bandpass filter that is printed on the circuit board; its purpose is to remove any unwanted signals before the receive signal is demodulated. A lowpass filter follows the amplifier to remove any harmonic spurious from the demodulation chain; this filter 1 s also printed on the circuit board. The active device used here is the same as the active devices in the Rx chain on the antenna.

Receive First Mixer and LO Tripler

The amplified receive signal is downconverted to an intermediate frequency of 446MHz in the first mixer. The mixer is driven by an LO signal at a level of about +9dBm.

The lower sideband output from the mixer is then filtered and sent on to the first IF AGC amplifier chain.

Receive First LO

The first LO signal is generated by amplifying the third harmonic of the transmitter LO signal, sent by coaxial cable from the transmitter module. The transmitter LO frequency at 2082MHz is bandpass filtered and amplified and the third harmonic is picked off by the bandpass filter (at 6246MHz) following the tripler. This signal is then amplified and filtered.

Receive AGC Amplifier and Gain Chain

The first IF AGC amplifier is controlled by the magnitude of the baseband signal sent on to the analog to digital converter. The purpose of the AGC signal is to keep the baseband signal at a level of about 800mV into the ADC. This function is implemented with a cascade of two amplifiers driven by the same control voltage. The ADC amplifier is followed by a bandpass filter and gain block to ensure enough signal is present in the demodulator chain.

Receive Second Mixer

The first IF amplified signal is downconverted to the second IF in the receive second mixer. This mixer is driven by an LO signal at 616MHz at a level of about +7dBm. The lower sideband of the resulting signal is filtered and sent to the second IF demodulator chain.

Receive Second LO

The receive second LO at 616MHz provides a carrier for the downconversion of the receive signal to the second IF at 170MHz. It consists of a commercial voltage-controlled oscillator, a buffer amplifier and a programmable synthesizer. The synthesizer is programmed by the microprocessor and returns its lock-detect status. The nominal output level for this LO is +7dBm.

Receiver Carrier Recovery

The receiver demodulator implements a data-directed carrier recovery loop that is ultimately used in the baseband downconverter. This loop performs two basic functions: it tracks the phase transitions generated in the tag and it also tracks the variation in phase of the received signal due to tag movement and tag crystal offset relative to, the cell controller reference signal.

Receive Baseband Downconverter

The second IF signal is downconverted to baseband at the baseband mixer. The LO for this mixer is derived from the receiver carrier recovery section and is at 170MHz at a level of about OdBm.

Receive Baseband AGC Signal Conditioner

The baseband signal is amplified and sent off to the digital board to be digitized in the analog to digital converter. A replica of this signal is averaged and used to set the gain of the AGC amplifiers in the first IF section. This signal is also sampled by another analog to digital converter (on command from the digital module) and sent to the digital module.

3D-1D Tag (T20 Edition)

General

Referring now to Figure 23, shown is an embodiment of a tag that may operate in the system of Figure 1 and Figure 25, that will be described further in paragraphs that follow. The tag is a low cost RF device which transponds radio signals from the cell controller transmit antenna to the cell controller receive antenna. Additionally, it modulates tag information such as the tag ID onto the signal sent to the cell controller receive antenna. The tag has three important design parameters: cost, battery life and size. Low tag cost is achieved by making the tag RF and digital design as simple as possible while using off the shelf components. Battery life is achieved with low-current RF and digital designs and by utilizing a small duty cycle. Size is achieved by a combination of surface-mount technology and careful antenna design. The tag is compliant with FCC Part 15.249 regulations. Figure 23 is a block diagram of the tag, showing all the major elements of the tag. Figure 24 is an overall specification of the RF elements of one embodiment of the tag of Figure 23. The following sections will describe the various elements of the tag in detail.

RF Input Section

The RF input section consists of four basic structures: an antenna 2402, an input bandpass filter 2404, a voltage variable attenuator 2406 and amplifier structures. The antenna element is external to the tag circuitry and for the T20 edition of the tag is a single patch having a gain of about 4.5dB. The input bandpass filter passes RF signals over a bandwidth of 80MHz centered at 2440MHz with a loss of less than 2dB and a delay nonlinearity of less than 4nsec peak to valley. This filter is nominally a three-pole response with at least 40dB of rejection in the tag output band of 5725-5815MHz. The amplifier structure consists of two gain stages, each having about 18dB of gain at 2442MHz.

Modulator and Mixer Section

This section consists of two basic structures: the biphase modulator 2408 and the mixer 2410. In the T20 edition of the tag, the biphase modulator is implemented with a pair of single-pole, double throw switches and transmission lines that differ in phase by 180 degrees. The mixer is implemented as a double-balanced diode quad. The sum and difference signals are present at the mixer output. The low frequency result (difference) is rejected at a highpass filter, so only the high frequency (sum) products are sent to the output amplifier.

Local Oscillator Section

This section consists of two basic structures: the phase-locked oscillator operating at half the desired frequency and the doubler. The phase-locked oscillator (at 1679MHz) consists of a VCO 2412 centered at the output frequency, a buffer amplifier, a standard fixed frequency PLL chip 2414, a passive loop filter and a fixed crystal. The PLL chip implements a ÷512 prescaler, a phase-frequency detector and a charge pump in a single package. The doubler is implemented as an amplifier followed by a filter.

RF Output Section

The RF output section is implemented with three basic structures: an amplifier, output filters and the transmit antenna 2416. The antenna element is external to the tag circuitry and for the T20 edition of the tag is a single patch having a gain of about 3dB. The output bandpass filter 2418 passes RF signals over a bandwidth of 100MHz centered at 5800MHz with a loss of less than 3dB and a delay nonlinearity of less than 4nsec peak to valley. This filter is nominally a three-pole response with ay least 40dB of rejection in the

tag input band of 2400-2484MHz. The amplifier structure consists of a single gain stage having about 18dB of gain at 5800MHz.

Microcontroller and Power Control Section

This section consists of two major elements: the microcontroller 2420 and the power control circuitry. The tag protocol requires the microcontroller to turn on the local oscillator section: long enough for it to stabilize fully and then to turn on the RF section. The loop bandwidth of the local oscillator is wide enough that the LO lock-time is a maximum of 0.75msec. There is also time required for the crystal oscillator in the PLL chip to stabilize: the sum of the lock-time and the crystal stabilization time is about 3.5msec. Once the LO is stable, during transmit the protocol requires that the microcontroller turn on the RF section and proceed to phase modulate the transmitted signal according to the requirements of the tag protocol. This protocol requires a tag bit time of about 19 $\mu$sec and an overall tag transmit time of about 2.3msec.

In the on state, the microcontroller samples the tag low battery indicator and latches this onto the tag housekeeping data bits. These bits form part of the tag datagram sent during transmit. After the tag ID, tag data bits and checksum have been modulated onto the transmit signal, the microcontroller then shuts off the RF section followed by the LO section. Finally, the microcontroller determines the off time (typically 5 seconds), loads the off time counter and puts itself into a very low-power state in which it simply counts down the off time counter. When the counter expires, the microcontroller wakes up and repeats the on cycle.

Tag RF System Description

This section describes in detail the operation of the RF elements of the tag, following the general outline given above. It consists of detailed descriptions of the RF input section, the modulator and mixer section, the local oscillator section' and the RF output section.

RF Input Section

The RF input section describes all the elements from the antenna to the modulator. This section consists of the Rx antenna, the input bandpass filter (Rx BPF), the voltage~variable attenuator and the low-noise amplifier (LNA). All of the elements of this system have a nominal impedance of 50 ohms and are connected to one another with 50 ohm transmission lines. The purpose of the input section is to make sure the incoming signal (nominally a spread-spectrumsignal centered at 2.442GHz) provides enough drive for the mixer and RF output sections without driving them into non-linearity that would violate the requirements of 15.249.

To this end, the nominal tag gain of 35 dB is adjusted in the RF input section to provide a constant, linear level using a hardware implementation of automatic gain control. The AGC is implemented with a detector diode at the RF output stage, which provides a DC voltage that is proportional to the RF output power. This DC voltage is amplified and smoothed in the AGC amplifier and then delivered to a combination of two voltage-variable attenuators: one GaAs MMIC in front of the LNA and a PIN diode attenuator, following the LNA. This combination is designed to have 35-40dB of attenuation range. This range is designed to allow tag linear operation from about 1 meter away from a cell controller antenna to the maximum range.

For tag ranges inside the 1 meter from a cell controller antenna, the circuitry associated with the LNA shutdown, controller monitors the AGC voltage. When this voltage reaches a value corresponding to the tag RF output of about 3dB below the point where the spectral re-growth of the desired spread-spectrum signal reaches the limits of 15.249 at the band edges, the amplifier output transitions, signaling the microcontroller that the low-noise amplifier needs to be shut down. This signal generates an interrupt in the microcontroller that forces the assertion of the OFF line to a switch, which removes power from the LNA bias line. This effectively adds 40dB of signal attenuation into the RF input chain. The hardware AGC then readjusts the voltage-variable attenuators to provide the best tag RF output level possible consistent with linear operation (now is back at the bottom end of the hardware AGC range).

Rx Antenna

The function of the receive antenna is to capture radiated signals at the desired input frequency over a spatial range as close to half-plane omnidirectional as possible. The receive antenna is a low-cost quarter-wave shorted patch antenna constructed of sheet metal. The antenna structure uses air as a dielectric and operates over the ground plane on the backside of the tag printed circuit board. The antenna makes connection to the rest of the RF input section with a simple RF feedthrough pin. The antenna has a midband gain of 4.6dBi and is linearly polarized.

Rx Input Bandpass Filter

The function of the input bandpass filter is to pass only the input signals of interest and to suppress signals that are not part of the MD system. The input receive bandpass filter

is a 3-pole monolithic ceramic dielectric resonator bandpass filter. It has a nominal insertion loss of 2.3dB and a nominal 3dB bandwidth of ~100MHz.

Voltage-Variable Attenuator

The function of the WA is to implement the AGC scheme described above. The voltage-variable attenuator is implemented as a combination of two elements: a monolithic microwave integrated circuit and a shunt PIN diode attenuator. The first element is placed in front of the LNA and has an effective dynamic range of about 33dB. The second element is placed after the LNA and has an effective dynamic range of about 7dB. The combination of the two on either side of the LNA results in a control range of about 40dB.

Low-Noise Amplifier

The low-noise amplifier is a GaAs monolithic amplifier. It exhibits about 20dB of gain with a noise figure of about 3dB. It serves to set the tag noise figure and to provide enough gain at maximum range for suitable tag operation.

Modulator and Mixer Section

The modulator and mixer section provides two of the essential system functions for a 3D-iD tag: the ability to identify a tag and the conversion of the input frequency to the output frequency. The modulator input comes from the microcontroller and is the digital representation of the tag datagram that the tag sends to the cell controller. The mixer inputs come from the RF input section and from the LO section and its output goes to the RF output section. Each of the elements of this system has a nominal impedance of 50 ohms and is connected to one another with 50 ohm transmission lines.

Modulator

The function of the modulator is to change the representation of the tag datagram from a digital signal to a biphase modulated RF signal. The modulator is implemented as a pair of GaAs non-reflective, single-pole, double throw switches that switch between two transmission lines whose difference in length at the tag receive center frequency (2.442GHz) is 180°. The particular switches chosen require differential drive, so a CMOS inverter is used to derive a pair of differential drive signals from a single digital control from the microcontroller.

Mixer

The function of the mixer is to translate the incoming 2.442GHz signal to the proper output center frequency (5.800GHz). The mixer is implemented a commercial double-balanced mixer. The input local oscillator signal to the mixer is at 3.358GHz and a filter in the RF output section passes only the sum signals and suppresses the difference signals from the mixer.

Local Oscillator Section

The local oscillator simply supplies the single tone used in the mixer to upconvert the incoming 2.442GHz signal to a 5.800GHz frequency. The oscillator is implemented as a phase-locked oscillator derived from the output of a voltage-controlled oscillator stabilized by a phase-locked loop. The final output of the local oscillator is 3.358GHz. This is obtained by filtering the second harmonic of the phase-locked oscillator that is running at 1.679GHz. The phase-locked loop is implemented in a standard CMOS RF chip and the loop filter is implemented as a 4-pole passive filter. The reference signal for the PLL comes from an onchip oscillator that uses an external crystal at 6.5MHz.

The VCO output level is at about -5dBm for the fundamental signal. An amplifier whose gain is about 13dB and whose output feeds both the PLL chip and the next amplifier stage buffers this output signal. The final amplifier stage is tuned to reject the fundamental signal and amplify the second harmonic at 3.358GHz to about -8dBm. A bandpass filter to remove everything but the desired second harmonic follows the final amplifier.

RF Output Section

The RF output section simply provides the proper radiated output signal for proper tag operation. The output section consists of a cleanup bandpass filter following the mixer, an amplifier, a final bandpass filter and the Tx antenna.

Tx Bandpass filters

The bandpass filters are identical 3-pole hairpin filters having about 3dB of insertion loss and a 3dB bandwidth of about 300MHz centered on 5.8GHz. The filters are implemented directly on the printed circuit substrate.

Tx Final Amplifier

The amplifier is a single monolithic microwave integrated circuit having a gain of about 18dB and a third-order intercept point of about +15dBm so that the output signal can be as linear as possible at the desired nominal output level of -7dBm.

Tx Antenna

The function of the transmit antenna is to emit radiated signals at the desired output frequency over a spatial range as close to half-plane omnidirectional as possible. The transmit antenna is a low-cost quarter-wave shorted patch antenna constructed of sheet

metal. The antenna structure uses air as a dielectric and operates over the ground plane on the backside of the tag printed circuit board. The antenna makes connection to the rest of the RF input section with a simple RF feedthrough pin. The antenna has a midband gain of 3.0dBi and is linearly polarized.

Microcontroller Section

The microcontroller section operates the protocol for the tag. It accepts user information and combines this with information about the tag state (such as battery voltage is low). The microcontroller then provides this digital stream to the biphase modulator. The microcontroller also monitors the state of the AGC shutdown input and commands the LNA bias off at the properties. It also directs the sequence of RF element power at the beginning of a datagram (LO on first, check for phaselock, then RF on) and controls the off-time under software direction. The processor controls DC power with FET switches.

Microcontroller

The microcontroller is a PIC16F84 CMOS low-power, general purpose digital processor. This processor derives all timing from a 4MHz ceramic resonator that establishes the internal clock stream. The internal clock is derived from the ceramic resonator frequency by dividing it by 4: thus, the internal clock operates at 1 MHz.

DC switches

The switches accept logic inputs from the microcontroller and turn DC power on and off to various system elements. One controls DC power to the LO section and to the majority of the RF elements. Another controls DC power to the LNA section, implementing level 2 AGC.

As in previously described embodiments included herein, an example embodiment of a single RFID system embodiment is shown in Figure 25, which is similar to Figure 1. Figure 25 shows a 3D-ID Cell Controller, which is a sophisticated digital radio that controls a set of antennas that cover a cell within a building. In an open indoor environment, each cell may be configured to cover about 10,000 square meters.

Like GPS satellites, 3D-ID Cell Controllers emit direct sequence spread spectrum signals that are received by the tags. Unlike GPS, 3D-ID tags do not include sophisticated circuitry and software to decode this signal; instead, they simply change the signal's frequency and transpond it back to the Cell Controller. Tag ID information is phase modulated onto the return signal. The Cell Controller extracts the Tag ID from this return signal, and also determines the tag's distance from the antenna by measuring the round trip time of flight of the radio signal. Advanced numerical filtering provides approximately one meter distance measurement accuracy.

Up to 16 antennas can be attached to a single Cell Controller via coaxial cable. (For simplicity, 4 antennas are shown in Figure 25). Both power and radio signals travel through this single cable, so no additional power source is necessary for the antenna. Both the Cell Controller and the tag are carefully designed to comply with FCC Part 15 regulations, so no license is needed for operation of this embodiment.

One version of the active tag emits ½ milliwatt of radio energy, enabling detection of the tag at a distance of 50 meters. This allows a tag about the size of a PCMCIA card to operate for over a year with a small battery. The tag transmits a "low battery" signal well before replacement is required. For applications where battery life isn't critical, higher power tags can support substantially longer range.

The Cell Controller quickly cycles among antennas, determining the distance between all of its antennas and a given tag. Once the distance to 3 antennas is found, the

tag's location in space can be estimated. In many situations, it's possible to get a good estimate of tag location from fewer than 3 antennas. For example, in a warehouse an aisle between racks can be covered by two antennas — one at each end. Most hallways can be similarly covered.

Tags '*wake up*' spontaneously, '*chirp*' their unique codes, and then go back to '*sleep*'. Each chirp takes 2.3 milliseconds. The chirping interval can vary based on application requirements. For example, tags attached to personnel might chirp every two seconds, while tags attached to inventory might be set to chirp once per minute. For infrequently used physical assets, special tags with motion detectors can be provided, that chirp infrequently when stationary and more frequently when in motion.

The tag data protocol includes a capability to report information provided by a closely integrated device. The types of data that might be reported through a tag are limited only by the creativity of the customer. For example, a specialized tag on a guard could transmit pulse rate, and a "panic button" could also be included. More generally, tag electronics could be used as an inexpensive RF stage for a handheld data collection device. Future versions of the tag will support two-way communication, implementing a low-cost form of "wireless LAN." Thus, this 3D-ID embodiment is not only a system to track people and assets, it also provides a cost-effective infrastructure to collect a wide variety of data throughout the enterprise.

The 3D-ID embodiment was designed with careful attention to minimizing cabling and installation costs of the Cell Controllers. The link from a Cell Controller to its antennas requires the installation of dedicated coaxial cable. Both power and RF signals are transmitted to the antenna across this cable, so no additional power source is necessary. This may be helpful for a situation where no power source is available close to the desired antenna location.

Cell Controllers, and by extension their tags, are designed as network devices. Inside the Cell Controller, an embedded version of Unix supports communication with the corporate network via TCP/IP. An open application programming interface (API) enables an application developer to extract data from the cell controllers using a publish/subscribe model via TCP/IP.

The 3D-ID system bring the benefits of GPS indoors, in a package that is accessible to a wide range of businesses. It also provides a cost-effective wireless infrastructure to collect a variety of data throughout the enterprise.

Now, what is described are the software aspects that may be used in conjunction with embodiments of the previously described hardware to process, collect, and use the data as may be gathered by the cell controllers.

Referring now to Figure 27, shown is an example of a system 2810 displaying the various services that may be included in one embodiment of a location system. To be described in paragraphs that follow are how the various services depicted in Figure 27 in system 2810 work with one another in providing tag data to various subscribers. Shown in Figure 27 is the configurator service 2812 with the configuration database 2814. Also shown are cell controllers 2816a-2816c, as previously described in conjunction with other figures, which in the system 2810, as will be described in paragraphs that follow, transmit data to various services within the system 2810. Also shown in the system 2810 of Figure 27 is the tadder service 2815, locator service 2820, the alerter service 2822, the recorder service 2824 connected to a recorder database 2828, and a broker service 2826. Each of the forgoing components will now be generally described.

The configurator service 2812 generally retrieves configuration information as used by services and applications. For example, it includes and provides tag data and various rules for determining and publishing tag data. The broker service 2826 generally serves as a go-between for transmitting data between various components of the system 2810. The data used by the configurator service 2812 is stored in the configuration database 2814. The tadder service 2818 s generally consolidates cell controller data. As is described in paragraphs that follow, the cell controllers transmit data which is collected and stored and organized by the tadder service 2818. The locator service 2820 is generally responsible for publishing tag location information to various subscribers. The locator service generates location information about tags as well as tag information about a particular location. This will be described in more detail in paragraphs that follow. The alerter service 2822 subscribes to the locator and publishes various messages in accordance with predefined conditions. The recorder service 2824 subscribes to data from other services saves it in the recorder database 2828 creating historical records that may be used by other applications.

Generally, information or data which flows in the system 2810 between the various services and databases previously described in accordance with a subscriber publication model. An application that needs data subscribes and is called the subscriber. An application that provides data and publishes the data is accordingly called the publisher. A subscriber is a process, for example, which obtains data from a publisher. This is done generally by subscribing to data via a subscription. Data is published by the publisher to the subscriber as requests in a subscription. A publisher is a process that publishes data to subscribers. When data is published, the publisher transmits the data to all interested subscribers. An interested subscriber is one that has an active subscription as indicated in a publisher subscription list. A subscription is generally a request sent by a subscriber to a

publisher. The subscription indicates to the publisher the information the subscriber is interested in receiving. A subscription list is an active list of all active subscriptions as maintained by the publishers, for example. Each time data is ready for publication, the subscription list is consulted and appropriate subscribers are accordingly sent data. When a subscription expires, it is removed from the list. Similarly, when new subscriptions or subscribers arrive, they are accordingly added to the subscription list.

The broker service 2826 is a service that directs subscribers to publishers. Generally, it is used to provide such functions as load balancing, fault tolerance, and scalability in the various publish/subscribe mechanisms which will be described in paragraphs that follow.

It should be noted that the foregoing functions provided by the various services, such as the broker service, may vary with embodiment. As known to those skilled in the art, the broker and other services described herein may be implemented using a variety of techniques. For example, the services in one embodiment may be implemented in a commercially available standard programming language, such as the "C" programming language, interfacing with standard Windows programming APIs.

The broker is the target for all subscriptions. When a subscriptions arrives, the broker consults the publisher list to determine which publishers publish the information requested in a subscription. The broker responds to the subscriber with a redirect messages which causes the subscriber to reissue the subscription request to the publisher as chosen by the broker. The publisher list is maintain by the broker and is a dynamic list of all active publishers and their information which is published. As the publisher comes online, it

registers with the broker. The broker accordingly updates its list of publishers with various publishers as they are added and deleted in the system 2810 of Figure 27. The pending list as will also be described in paragraph that follow is also maintained by the broker service 2 826 and it is a dynamic list of subscription requests waiting for the appropriate publisher. The pending list, for example, may be a list of subscription requests when a subscriber requests information where there is currently no publisher of that information. When a publisher comes online able to fulfill one of the requests, that request is filled and the broker sends the subscriber a redirect message to allow the subscriber to obtain the necessary data from the publisher.

The tadder service 2818 is responsible for receiving and organizing tag antenna distance information (TAD) information as may be send by a cell controller 2816a-2816c. A cell controller generates TAD data which includes information such as the tag identifier, the antenna that received the signal from the tag and the distance from the antenna that the tag signal was received. The locator service 2820 is responsible for transmitting and processing data related and location of a tag referred to as LOC data. In the system of 2810, locations of the tag are defined by a location ruler set. The locator service 2820 uses the location rule set as may be stored and managed by the configuration service 2812 and configuration database 2814, and the TAD information to generate location or LOC data. Generally, the location rules that may be included in the configuration database 2814 uses rules to define locations in terms of TAD data.

As an example of the relationship between TAD and LOC data, the rule CC1.1=24 states that if the tag is within 24 feet of the first antenna of the cell controller name CC1,

then the tag is in the location which the rule defines. A location rule set is made up of many locations with their associated rules organized in hierarchal fashion.

The alerter service 2822 publishes data when an alert conditions occurs. For example, in a system 2810 an alert condition may occur when a tag enters or exits a predetermined area or when a tag remains stationary for a set period of time. An alert may also be generated when one tags enters an area without a second specific tag, or when two tags are near one another, or two tags are far apart from one another. Generally the alerter service 2822 generates the alert data by applying the location data it subscribes to and its alert rule set.

At system startup time in one embodiment, the initial elements may be the configurator service 2812, the configuration database 2814 and the broker service 2826. The cell controllers are next recognized and added as part of the system of 2810 and are accordingly transmitted or sent configuration information by the configurator service 2812. The tadder service 2828 comes online next and accordingly queries that configuration for appropriate setup information. For example, the tadder 2818 may request a specific IP and port ID of a broker and a list of TAD publishers it should subscribe to. Generally, this type of information such as the IP and port ID of the broker and the list of TAD subscribers is stored in the configuration database 2814 and managed by the configuration service 2812. Additional types of data may be stored in the recorder database and the configuration database 2814 will be described in conjunction with other figures as needed with explanations.

In accordance with the set of information received, the tadder service 2818 subscribes to its assigned publishers via the brokers service 2826. In this particular embodiment, it subscribes to the cell controllers 2816a-2816c. This is shown by the transmission of the TAD data from each of the cell controllers 2816a-c to the tadder service 2818 in Figure 27. A similar process is repeated for the locator service 2820. For example, the locator service 2820 receives setup information including publishers and a location rule set from the configuration. The locator service 2820 accordingly subscribes to the tadder service 2818 via the broker service 2826. The tadder service 2818 subsequently publishes TAD data to the locator service 2820. Subsequently a similar process is repeated for the alerter service 2822.. The alerter service, for example, receives setup information including publishers subscribed to as well as an alert rule set from the configuration service 2812. Subsequently, the alerter service subscribes to the locator service 2820 via the broker service 2826. Accordingly, the locator service publishes LOC data to the alerter service 2822. The recorder service 2824 subsequently comes on line and a similar process is also repeated as for the alerter service, the locator service and the tadder service previously described. The recorder service, for example, receives setup information from the configuration service 2812. The recorder service subsequently subscribes to the alerter and location services via the broker service 2826. The locator and alerter services publish data to the recorder service 2824. Data as received by the recorder service 2824 is stored in the recorder database 2828.

As previously described, the publish/subscribe mechanism is used to facilitate communication between the various services and applications and services in the system 2810. What was previously described is the general description of what system components are subscribers and publishers of the various types of data in the system 2810. What will

now be described in more detail are the typical publish and subscribe interactions between the various components of the system 2810.

Referring now to Figure 28, shown are the five components as may be included in the system 2810 of Figure 27 which are involved in the publish subscribe mechanisms as previously described. For example, as will be described in paragraphs that follow regarding the publish/ subscribe mechanism, the recorder service is a subscriber of alert data as published by the alerter service. Similarly, the tadder service is a subscriber of TAD data as published by each of the cell controllers.

Figure 28 includes a TAD subscriber 2840, a LOC subscriber 2842, a broker 2844, a TAD publisher 2846, and a LOC publisher 2848. Each of the foregoing elements are subscribers or publishers of the various type of TAD and LOC data in accordance with the naming of the elements. A broker 2844 as previously described is included in the broker service 2826 and serves as a go-between for the subscription and publication of data between a subscriber and a publisher in this model. Generally, each of the subscribers such as the LOC subscriber 2842 and the TAD subscriber 2840 are clients or client processes in the system 2810 as included in Figure 27.

As will be described in conjunction with Figures 29 through 36 in paragraphs that follow, shown is a protocol by which TAD data is communicated or published between a TAD publisher and a TAD subscriber as facilitated by the broker. It should be noted that the techniques that will be described may also be applied in a similar fashion to LOC data publication and subscription.

Referring now to Figure 29, shown is the broker 2844 communicating with the TAD publisher 2846. Upon a TAD publisher process starting up, the TAD publisher 2846 registers with the broker 2844 as a publisher of certain kinds of data. In this particular instance, the publisher 2846 publishes TAD data. Referring now to Figure 30, shown as the next step in the protocol where the broker 2844 adds the publisher 2846 to its publisher list of TAD data.

Referring now to Figure 31, a TAD subscriber 2840 starts and sends a subscription request for TAD data to the broker 2844. Referring now to Figure 32, the broker 2844 consults its publisher list to see if a publisher of TAD data exists. In this particular example, the broker 2844 finds such a publisher at a particular network location, for example, at IP 1.2.3.4 having a port ID of 5003.

Referring now to Figure 33, the broker subsequently redirects the subscriber to the TAD publisher at the network location as determined by the publisher list. Referring now to Figure 34, the subscriber resubmits its request for the subscription directly to the TAD publisher. Referring now to Figure 35, the TAD publisher 2846 responds to the subscription request by transmitting a subscription accepted message to the TAD subscriber 2840. In Figure 36, both the TAD subscriber 2840 and the TAD publisher 2846 make note of the subscription. For example, the TAD publisher stores the IP and port ID that the subscriber will be listening to as the network location transmitted by the broker. Similarly, the subscriber makes note of the subscription in a subscription list as maintained by the subscriber 2840 indicating those TAD publishers which the subscriber 2840 subscribes to.

Referring now to Figure 37, shown is a flowchart summarizing the method steps in the previously described embodiment for how information is transmitted between a TAD publisher and a TAD subscriber and facilitated by the broker in a system 2810 of Figure 27.

Referring now to step 2860, the TAD publisher initially registers with the broker. At step 2862, the broker adds the TAD publisher to its publisher list. At step 2864, the TAD subscriber sends a subscription request for TAD data to the broker. At step 2866, the broker consults its publishing list to see if TAD publisher exists for the requested information in the subscription. At step 2868, after the broker determines that such a TAD subscriber exists, the broker redirects the TAD subscriber to the TAD publisher by transmitting TAD publisher network location information to the TAD subscriber. At step 2870, the TAD subscriber submits its request directly to the TAD publisher at the network location as transmitted by the broker at step 2868. At step 2872, the TAD publisher responds with a subscription accepted message. The subscription accepted message is sent from the TAD publisher to the TAD subscriber. At step 2874, the TAD subscriber and the TAD publisher both make note of the subscription as maintained locally.

The method steps summarized in Figure 37, as well as the steps depicted by Figures 29 through 36 in establishing a publisher subscriber communication means, may also apply to a LOC subscriber and LOC publisher in the manner similarly described with a TAD publisher and a TAD subscriber.

What will be described in paragraphs that follow in conjunction with Figures 38 through 46 are method steps for establishing a subscriber publisher relationship for communicating data prior to the publisher of the requested subscription being registered and

on line in the system 2810 of Figure 27. It should be noted that this method which will be described uses a LOC subscriber and LOC publisher. However, similarly as previously described in conjunction with the TAD subscriber and TAD publisher, the method that will be described is not specific to a LOC publisher and LOC subscriber. The same method steps that will be described in conjunction with Figures 38 through 46 also apply similarly to a TAD subscriber and TAD publisher.

Referring now to Figure 38, the subscriber starts up and issues a subscription request for LOC data to the broker 2844. Referring now to Figure 38, the broker 2844 as included in the broker service 2826 of Figure 27, checks its local publisher list. It determines that there is no publisher of LOC information as requested by the LOC subscriber 2842. Accordingly, the broker 2844 stores the subscription request for LOC data by the LOC subscriber 2842 in its pending list. Referring now to Figure 40, the LOC publisher 2848 starts and registers with the broker 2844. Referring now to Figure 40, the broker accordingly adds the LOC publisher 2848 to its publisher list. Referring now to Figure 42, the broker subsequently checks its pending subscription list. The broker searches the pending list and notes the LOC subscribers request matching the newly-added LOC publisher 2848. Accordingly, the broker 2844 issues a redirect message to the LOC subscriber 2842 with the network location of the LOC publisher 2848.

Referring now to Figure 43, the broker accordingly removes the subscription request from its pending list. In Figure 44, the LOC subscriber 2842 resubmits its subscription request directly to the LOC publisher 2848. This is done in a manner similarly as previously described in conjunction with the TAD publisher and TAD subscriber without having to go through the broker using a pending list request. Referring now to Figure 45,

the LOC publisher responds to the subscription accepted message to the LOC subscriber. Referring now to Figure 46, both the LOC publisher and the LOC subscriber accordingly note the subscription and manner similar to the TAD subscriber and TAD publisher as previously described.

Referring now to Figure 47, shown is a flowchart summarizing the method steps previously described in conjunction with Figures 38 through 46 of what happens in one method of establishing a publisher subscriber data flow when the subscription request is made prior to the publisher being online.

In step 2880, the LOC subscriber sends a subscription request for LOC data to the broker. At step 2882, the broker consults its publisher list to see if the LOC publisher exists and adds the subscription request to its pending list since it has determined that no such LOC publisher is currently online. At step 2884, the LOC publisher starts and registers with the broker. At step 2886, the broker adds the LOC publisher to its publisher list. At step 2888, the broker checks its pending list in response to the new publisher coming online and identifies the LOC subscriber's request. Accordingly, the broker redirects the LOC subscriber to the LOC publisher by transmitting the LOC publisher network location to the LOC subscriber. At step 2890, the broker updates its pending list to remove the LOC subscription request. At step 2892, the LOC subscriber submits its subscription request directly to the LOC publisher at the network location as transmitted by the broker in step 2888. At step 2894, the LOC publisher responds with a subscription accepted message. At step 2896, the LOC subscriber and LOC publisher note the subscription.

It should be noted that the steps described in conjunction with Figure 47 may be performed similarly by a TAD publisher and a TAD subscriber when obtaining TAD data service when a TAD subscription request is issued prior to a TAD publisher being online and available as indicated by the broker publisher list.

Prior to a publisher, such as a TAD publisher or a LOC publisher, participating in performing the previously described method steps in the publish/subscribe mechanism, a publisher performs certain tasks as will be described in the paragraphs that follow in connection the following figures.

Referring now to Figure 48, generally, when a publisher comes online prior to participating in the publisher's description process, a publisher requests information from the configurator 2900. Information requested by a publisher 2902 may include, for example, rule sets, port and other network identifiers, and the like. Generally, information as furnished by the configurator 2900 includes configuration information related to the configuration of the system 2810 in Figure 27. Subsequently, a publisher 2902 registers with a broker 2904. This, for example, was previously described in conjunction with processing steps as performed between a subscriber and a publisher. If the publisher 2902 obtains data from other publishers such as 2906 and 2908, the publisher 2902 issues subscription requests to its assigned publishers as indicated by the configurator 2900. Generally, a publisher may obtain data from one or more other publishers as shown in Figure 48 for purposes such as data propagation. In other words, a subscriber of publisher 2902 receives data that has been propagated through publisher 2902. This data that is replicated may be data that is published by, for example, publisher 2906 and publisher

2908. This technique, for example, may be used to propagate data throughout the system 2810 of Figure 27.

The foregoing steps just described may be performed prior to a publisher 2902 becoming available, for example, to service requests. Once the publisher 2902 has been started as a process, and that it is available to publish data, the publisher 2902 listens for requests for subscriptions, such as from clients 2910.

Generally, in one embodiment, a publisher may listen for subscription requests such as by using a busy wait loop or using operating system services for data communication forwarding client requests, for example, as an interrupt or data exception to the publisher process 2902. In one embodiment, this may be implemented using TCP/IP sockets that may be included in an operating system, such as Windows NT (host) and Linux (CC).

Once a publisher process is on line, data is published to all of its subscribers when it has appropriate data for those subscribers. For example, referring back to Figure 27, when the tadder service 2820 receives data as by the cell controllers 2816A through 2816C, the tadder service publishes data to various subscribers including the locator service 2820 and the recorder service 2824.

Additionally, when the publisher is available online, it must listen for other types of information requests, for example, regarding the status of the publishing information, its current state, copies of its subscription lists, etc. as may be requested by various system service components in accordance with the different facilities available for use.

Referring now to Figure 49, shown is an example of an embodiment illustrating the various interservice relationships, the responsibilities of a client or a subscriber in the system 2810 of Figure 27. Generally, as will be described in paragraphs that follow, for a subscriber or a client to participate in the previously described published subscription mechanisms, it must perform the tasks as will be described in paragraphs that follow. Initially, a subscriber 2912 determines the configurators 2900 location. This may be done, for example, by having a subscriber 2912 listen for broadcasted information, such as by a multi-cast of a message by the configurator 2900 in which the configurator 2900 broadcasts its address on the network to as system components for example. Additionally, a subscriber such as 2912 may obtain the location for the configurator 2900 by using a preassigned IP and port ID identifying the configurator 2900. Subsequently, this subscriber 2912 requests information from the configurator 2900. This information may include, for example, a list of tag locations and alerts, as well as publisher and broker information. This information, for example, may be used by the subscriber and client in processing tag location, data and information.

The foregoing description has been a functional description of the various services and the relationships between them as included in the system 2810 of Figure 27. Now what will be described in paragraphs that follow are various hardware embodiments with the particular software services included on various hardware configurations. It should be noted that in the foregoing descriptions associated with the figures that follow, the hardware and software combinations are merely exemplary embodiments not to be construed as limiting the principals of the invention.

Referring now to Figure 50, shown is an example of an embodiment of a hardware configuration for the system 2810. Generally, the embodiment of Figure 50 includes a server 2926, a cell controller 2924, an associated antennae 2922a-2922d. The server 2926 and cell controller 2924 communicate over communication medium network 2928 with client systems 2920a-2920dD. The server 2926 may be a Windows NT server. Installed on the server 2926 in this embodiment is the Internet Information Service (IIS) and various services. In this embodiment, the IIS may be used if a client application interacts with or uses a browser.

The cell controllers, such as 2924 in this embodiment have access to a Dynamic Host Control Protocol (DHCP) server. Additionally, IP adjustments are reserved in this embodiment for the various cell controllers on the DHCP server. The client system, such as 2920a-2920d may be running an operating system such as Windows 95, 98 or Windows NT. Generally, these clients 2920a-2920d may also be web clients running, for example, Netscape 3.0 or greater or running IE 3.0 or greater. This will be described in paragraphs that follow, on client systems 2920a-2920d may not reside on the same subnet as the cell controller 2924 and the various 3D-iD servers such as 2926.

The embodiment of Figure 50 is able to carry out and implement the functions previously described. However, there may be network trafficking issues, as well as a problem with scalability.

Referring now to Figure 51, shown is another hardware embodiment of the system 2810 where the server responsibilities, as previously performed by 2926 of Figure 50, are divided between three server systems 2940 of Figure 51. In the embodiment of Figure 51,

the possibility of servers sluggishness or overload may be overcome. Contrasting Figure 50 in which all the services were shown as being allocated to the server 2926, the various server services such as the tadder locator and alerter services are hosted on their dedicated systems in 2940. Additionally, on the NT server 2926 shown are various 3D-iD services, such as the configurator, the broker, the recorder and the time server.

In this embodiment, it should be noted that the time server may be used to synchronize cell controller clocks with a host computer system clock, as may be included in the server. Note that various messages, such as TAD messages (see Figure 60), may be time stamped. Time server provides for keeping this timestamps consistent across the network of the system.

It should also be noted that when determining a system configuration for load balancing to ensure that the system will run efficiently, it may be necessary to take into account the load balancing in accordance with the various server services provided. This load balancing may be performed by configuring the system, or with a dynamic load balancing technique known to those skilled in the art which may route data requests in accordance with system load parameters.

Referring now to Figure 52, shown is another hardware embodiment of an example of the hardware configuration for the system 2810 of Figure 27. In particular, introduced in Figure 52 hardware embodiment is the concept of a subnet. As shown in Figure 52, the subnet separates the various location data traffic from the general network traffic. This generally corrects for potential network traffic issues and allows for scalability as previously described as possible disadvantages with regard to the hardware embodiment

described in conjunction with Figure 50. It should be noted that in the embodiment of Figure 52, various server services are shown as being allocated on the NT hardware server 2926. The services include, for example, the configurator, the broker, the tadder, the alerter, the recorder, the locator and the time server, similar to those previously described in conjunction with Figure 50.

Referring now to Figure 53, shown is another hardware configuration of an example of a hardware embodiment of the system 2810 of Figure 27. Generally, the hardware configuration shown in Figure 53 combines load balancing with the use of a subnet. It implemented the load balancing technique as previously described in conjunction with Figure 51, for example, by dividing up the various server services provided on different server processors 2940. However, a subnet is used also in the embodiment of Figure 53 similar as previously described in conjunction with Figure 52.

Referring now to Figure 54, shown is another hardware embodiment of the system 2810 of Figure 27. In this hardware configuration, there is a multiple site configuration that generally includes subnets at each locator site or facility. Referring now to Figure 54, subnets 2970a-2970b each respectively are associated with a site used in gathering locator data by various cell controllers and their associated antennae. In the embodiment of Figure 54, the location data is divided into two functional groups, each of which is generating TAD, LOC and ALR data. For example, the site associated with subnet 2970A generates information which has been designated as Tad1, LOC1 and ALR1 data. Similarly, subnet 2970b associated with a second site generates data which has been designated as group association Tad2, LOC2 and ALR2 data. Each of these sites, as controlled by servers 2926a and 2926b, forward data to a central server 2966, which publishes compilations of these

data groups Tad1, LOC1, ALR1, Tad2, LOC2 and ALR2. The central server 2966 also stores the configuration database 2814 previously described in conjunction with Figure 27.

Client systems may access any of the servers 2966, 2926a or 2926b to provide the data needed. In other words, server 2966 publishes both Tad1 and Tad2 data. Server 2926a publishes only that data which is designated as Tad1. Similarly, server 2926b publishes that data which is designated as Tad2, or in this particular subnet. Thus, a subscriber requiring Tad1 subnet data may subscribed to server 2966 or server 2926a.

As will be described in conjunction with Figure 55, one of the functions an application program or client may perform is updating various types of data within the system. For example, an application or a client may update tag and LOC rule set data.

Accordingly, a mechanism or technique is needed within the system 2810 for updating the various stores of data as maintained in the various functional service components included in the system 2810 of Figure 27. The technique which will be described is referred to as a unicast messaging technique in which a message broadcast is made from a single sender to a single receiver. In the system 2810, the unicast technique is one method that may be used in the system to perform data updates systemwide. For example, in paragraphs that follow, a client application will update rule sets or tag status information. This data will subsequently be updated on all services and other client applications.

Referring now to Figure 55, shown is a flow diagram of the various messages communicated to perform data updates as made by the client application using what is

generally known herein as the unicast technique. In block 3000, the client application generates a unicast message. In this particular example, the client application may be executing on a client system which is a tag activator or the configurator application. Note that the tag activator is described in more detail in paragraphs that follow.

5

The client application 3000 sends a unicast message to the configurator service as indicated by block 3004. The configurator application accepts the unicast message and redirects or further forwards the unicast message to every service as indicated in the configurator database 2814. Thus, the configurator application as included in the configurator service 2812 of Figure 27 performs the lookup in the configurator database 2814. It should be noted that many different embodiments and data structures may be used in structuring the configurator database 2814. For example, one embodiment may implement this using Microsoft Access. The database content may be communicated to other services using messages.

15

In this particular example, the unicast message is further forwarded to tadder service 3006, locator service 3008 and other services 3010. These services accordingly process the unicast message, and redirected to each of their client applications currently having active subscriptions in accordance with the subscription list located with each of the services. In this particular example, each of the client applications is notified by the various services in accordance with the subscription lists maintained by each of the services are noted by elements 3012 through 3020 of Figure 55. This redirection or further propagation of the data in accordance with the subscription lists causes an update of each of the data stores and lists maintained by the various components of a system 2810 of Figure 27 using the unicast technique.

Previously described has been the various service components represented as functional boxes in Figure 27 system 2810. Also having been discussed are various hardware configurations and associated services as may execute on various client server configurations. What will now be described as the various workings of the hardware are the cell controller, antenna, and tag level. It should be noted that this hardware has been previously described in conjunction with embodiments such as Figure 1. This hardware will now be described in conjunction with the data that it generates.

Generally, the TAD data as previously described as tag antenna distance data, refers to information that the cell controller may generate. TAD data includes, for example, a unique tag identifier, the antennas which saw the tag, and the distance that the antennas are located from the tag.

Referring now to Figure 56, shown is an example of an embodiment of a cell controller, various antennas and tag which are associated accordingly with the cell controller. Generally, the tag such as 3034a and 3034b may be attached to various assets, objects and people, which are to be tracked. One or more antennas may be associated with a cell controller 3030. In this particular example, for antennas 3032a-3032d are associated with cell controller 3030. It should also be noted that the number of tags, number of cell controllers, and number of antennas may vary with each particular embodiment. In this particular system, each cell controller may be connected to up to 16 antennas, even though our example shown in Figure 56 only show four antennas. Further, the cell controller in this particular embodiment is connected to the four antennas 3032a-3032d via a coaxial cable. In this embodiment, the cell controller 3030 sends a 2.442 gigahertz radio signal to antenna 3032a via the coaxial cable connecting them. Antenna 3032a subsequently broadcasts this

signal to its surrounding area. The antennas such as 3032a listens for a 5.8 gigahertz return signal, such as from a tag. Without any tags, the return signal as previously described will simply be noise. The signal received by the antenna 3032a is further forwarded to the cell controller 3030, which processes the return signal. In the case of a tagless configuration, the cell controller processes and treats the return signal as noise. The process by which the cell controller operates has previously been described in conjunction with other figures. Consequently, the cell controller 3030 repeats this process for the remaining antennas 3032a-3032d. The cell controller 3030 continuously cycles through all of the antennas 3032a-3032d repeating the same process. The cell controller in this embodiment requires .8 ms to cycle through all sixteen antennas in an embodiment that includes sixteen antennas rather than four as shown in conjunction with Figure 56. Tags, such as 3034a and 3034b "chirp" at regular intervals, for example, ranging from .5 seconds to 5 minutes. In this particular embodiment, a tag chirp lasts for approximately 2.3 ms. When a tag is chirping, it is continuously listening to the 2.442 gigahertz frequency signal as transmitted by one of the antennas 3032a-3032d. The signal is received by a tag, such as 3034a, and modulates onto the signal the tag identifier or serial number. This resulting signal is rebroadcasted by the tag, such as 3034a, to the antennas as the return signal at 5.8 gigahertz in this embodiment.

Generally, while the cell controller 3030 cycles through the antennas, the tag chirps and rebroadcasts a return signal with the tag identifier as previously described, which is subsequently received by one of the antennas 3032a-3032d. Once an antenna cycle has been completed, the cell controller selects the signal from the antenna which receives the strongest tag response and uses this signal from which to determine the tag identifier. For each of the received signals, the cell controller determines a distance. The cell controller

uses the delay between the transmission of the 2.442 gigahertz signal and the reception of the 5.8 gigahertz signal to solve for the distance between the antennas and the tag using the method previously described in conjunction with other figures. This distance is referred to as the tag antenna distance or TAD data, which is then forwarded subsequently to the cell controller subscriber on the software side of the system. In other words, the strongest signal received is used to determine the tag identifier, and distances are associated with this tag identifier for each received signal from each antenna.

The system previously described may be used, for example, to track assets and personnel. The system may include various hardware and software components as previously described. Generally, the techniques described herein may be used to move data, as generated by the cell controllers, to end user applications which may be considered clients. Previously described is the data flow in the system between end user application and cell controllers which gather or produce the data.

Generally, the configurator service, or configurator, subscribes to the configuration database that includes configuration information. Accordingly, the configurator service communicates the information to services upon their startup. Additionally, the configurator responds to information or requests about the system infrastructure, and broadcasts update messages to notify services, for example, when an administrator performs a data update.

It should be noted that in one embodiment, the configurator stores persistent information in a database. The configuration service is the only service to directly access this database. Other services may obtain configurator database information using commands or messages. Commands used may be ASCII string commands as understood

and interpreted by the services. Commands may also be in accordance with predetermined database command formats, such as SQL and XML, and may vary with software included in each embodiment.

In one embodiment, a subscription from the tadder service for tadder information takes the form of "TAD, CCN", where CCN is a cell controller number. For example, a tadder subscription for "TAD, CC5" indicates that the tadder service is interested in all TAD data from cell controller 5. When the tadder service receives data from a cell controller, it accordingly forwards data to all of its subscribers. The tadder service may be viewed as the point of contact for TAD data information regardless of what cell controller generates the data.

The TAD view application is an end user application in this embodiment. It performs functions, such as displaying in real time all of the tag data readings. Whenever a cell controller generates a reading, for example, the TAD view application displays it on a display device. TAD view in this embodiment subscribes to all of the tag data in order to obtain the TAD readings that are to be displayed. Similarly, the locator service subscribes to the TAD service for all of the TAD information and publishes tag information to its subscribers. The locator may generate location information about any or all types, as well as tag information for a particular location.

The alerter service subscribes to the locator service for all LOC information, as well as tad information. Accordingly, the alerter generates certain alarm conditions such as entry, exit, stationary or other conditions in accordance with movement or positions of tags.

Alert data format is described in more detail in paragraphs that follow in conjunction with other figures.

Generally, the software components which function as software server processes include those basic components which the cell controllers use to publish data and which the client applications rely on for their data. In one embodiment, the components previously described include the configuration, the broker, the tadder, the locator and the alerter.

It should be noted that at startup, any component of the system may be initialized or started first as each is capable of starting in any order. It should generally be noted that when one of the components in the system of Figure 27 starts up and is initialized, there may be an initial contact with the configurator service to receive configuration information.

In one embodiment, the configurator service process is the first starting process. It is assigned a TCP/IP port number for handling incoming subscription requests. Similarly, other processes such as the broker process comes online and it is also assigned a TCP/IP port number for handling incoming subscription requests. It should be noted that the various processes, such as the broker process handling a large number of traffic, may be assigned more than one port number in accordance with the requirements and specifics of a particular embodiment and system.

Referring back to the system 2810 of Figure 27, what will now be described is one embodiment of how each of the components becomes part of the system of 2810. In other words, what will be described is the startup process in one embodiment for each of the

various components shown in system 2810 of Figure 27. It should be noted that other orders for initialization and incorporation into the system as an active member are possible.

Initially, all of the processes are components on the server system are initialized. This includes the following services and related processes included therein: the configurator, the broker, the tadder, the locator and the alerter. On the server, the configurator process is the first to begin. It is assigned a TCP/IP port address for handling incoming subscription requests. In this particular example, the TCP/IP port number is 5001. The broker process in the server begins next. In this particular example, it is assigned two port ID numbers, for example, TCP/IP port number 1072 and 1073 for handling incoming subscription requests. Subsequently, the broker subscribes to configuration parameters from the configuration. The configuration accordingly responds to the broker's initialization request and provides information to the broker. The tadder is initialized and is also assigned a port number for handling incoming requests and another port for handling incoming publication data. The tadder subscribes to configuration parameters via the configuration which responds with the information, including, for example, the cell controllers and broker IP address and port ID information. Generally, the tadder process subscribes to each of the cell controllers. It also maintains a list of cell controllers in the system. When data is received, the tadder process immediately redirects the data to all of its subscribers. Additionally, the tadder process needs to issue subscriptions of its own for all of the cell controller tad data as indicated in Figure 27. The tadder process consults its preconfigured list for the subscriptions.

In this particular example, the tadder prepares a subscription list for the first of its TAD data requirements. The tadder process has been configured in this example to send all

components previously described, which are the locator, alerter, recorder, tadder, broker and configuration services.

In particular, the system of Figure 57 illustrates how the data may be produced and managed by server 3056 and its components may be used by various client applications, such as 3052a and 3052b, and may be used also as an input to a database, such as database 3060. In particular, an API 3054 is provided for a communication mechanism by which a client application such as 3052A may obtain tag data and information from the server 3056. Similarly, a client application such as 3052 may use the API 3054 to obtain data from server 3056. Another possible use for the tag data as produced and managed by the server 3056 and the software components included therein is to use as the data feed for a database, such as database 3060. This database 3060, for example, may be an SQL or Oracle database. Such data may be stored, for example, by the recorder service of the server 3056. Generally, the access by which the database 3060 obtains the data from the server 3056 through making database calls and inquiries to store and retrieve data to and from respectively the database 3060. For example, the data flow provided between the server 3056 and the database 3060 may be ODBC compliant.

Various applications, such as client application 305a and 3052b, may be used to access and display data, for example, to a user in various different viewpoints. For example, a client application may be a viewer which displays various alert information or conditions. Additionally, an application may view data in a the form of a tree structure, such as displaying data by zone where a zone may correspond for example to one of the subnets previously shown in conjunction with other figures. Another type of application that may be used by a user to view data is a TAD view application to actually view an

subscription requests to a particular IP address and port number of the broker. Similarly, each of the other components or services of a system 2810 of Figure 27 are initialized and come on board to become a member of the system 2810.

At the point where all of the components of Figure 27 system 2810 are initialized and up and running, client applications subscribe to and may also use the information propagated and provided by the various services of the system 2810. For example, the TAD view, LOC view and ALT view applications as shown in Figure 55 of elements 3012 through elements 3020 are also initialized. It should be noted that these applications may be run on the various client workstations and other computer systems as shown in Figure 55. Generally, as with all other subscribers, the client applications may direct their subscriptions to the broker. Alternatively, they may direct their requests directly to a particular publisher rather than through a broker process.

The general process has been previously described as to how a client application as well as other subscribers of the system 2810 of Figure 27 are initialized and made an integral part of the data flow of a system.

Referring now to Figure 57, shown is an example of an embodiment of the location system and how it may be used by various applications and other software processes. Shown in Figure 57 are cell controllers 3050a-3050C. These cell controllers are similar to those previously described in conjunction with other figures. Also shown is server 3056. Generally, server 3056 is one as previously described, which may be, for example, a computer system which is a dedicated processor running, for example, the NT operating system, in particular, the server software. Included in server 3056 are those server

organization of TAD information as obtained by the system. It should also be noted that an embodiment of the invention which provides access for example to a database such as 3060 to data provided by the location system may also be compliant with other types of database accesses, such as using a standard SQL interface to provide access as needed to those databases that may require the SQL interface to obtain data from the information system and incorporated into the database of a particular type. In other words, rather than have an API such as 3054 which provides as the communication mechanism or data transfer means, various types of interfaces for different databases may also be provided such as SQL command interface or other equivalents thereof in accordance with the different types of databases as may be used with the server 3056 to store and organize data.

It should be noted a varying number of antennas and cell controllers provide different types of data coverage. Specifically, three types of coverage may be provided including portal coverage, zonal coverage, and full coverage. Generally, portal coverages addresses entrances and exits where what is generally kept track of that left a particular room, for example, or what floor a particular asset is on. Zonal coverage generally enables the division of the site into large zones. For example, zonal coverage addresses the requirement for ascertaining for what is in a particular wing or on a loading dock and who is in the lobby. Full coverage continually tracks any tag within plus 10 feet or 3 meters. Generally, portal coverage requires the least number of antennas while full coverage requires the greatest number.

The objective of the previously described hardware configuration is to produce tag antenna distances or TAD data as previously described. Once this TAD data has been generated it is subsequently forwarded to the viewpoint to the server such as servers 3056.

Generally an in user cannot use raw TAD data. Knowing the distance of an asset for one particular antenna may provide minimal use. Therefore, various services interpret the TAD data as included in the server that is generated by the hardware system. Through custom configure conversion tables the software services as included in the server convert multiple leads of TAD data from different antenna and different full controllers to what is called LOC data as previously described. Additionally, the software services as included in the server 3056 generate alert or ALR data as previously mentioned. Generally, as previously described, alert data is data based upon certain trigger events for a single tag or pairs of tags.

Generally, the service component as may be included in the server 3056 are processes implemented using various NT services and mechanisms to implement the publish/subscribe mechanism previously described. For example, in one embodiment a cell controller may include the Linux operating system.

Various client applications, such as client applications 3052a and 3052b, present data generated by the services to the user also previously described. Generally, the client applications interact with the services such as those on the server using the API 3054 to get information and subsequently display organized data. The role of the client application, for example, and the services they subscribe to for information may vary in accordance with the use and needs of the information generated by the system for location data.

The database 3060 may also include different types of data as produced by the system depending on the needs and data requirements of other components that interface with the database. For example, in one embodiment, the database 3060 may have a

database format that is ODBC compliant and also compatible for use with an Oracle and SQL server. Stored in the database may be ALR or alert data, TAD data, LOC data, and user and query audit information. Other combinations of data types are also possible in accordance with each particular embodiment.

In addition to various client applications that may be used by users of the system, there may also be included for administrative type applications as may be used to set up and manage the various services included in the server for example. An application which may function as a client that which may be used for administrative purposes includes the configuration application that may be used when the system is first set up and when the system is changed. Generally, this configuration application is used to carry out the process. It is not generally to configure client applications, but is rather to configure the various hardware and software options that may be included in the various hardware and associated software components as in the server 3056.

In one embodiment, there is an administrator interface that resides upon a computer system hosting the configurator service and related processes. Generally, this interface allows a system administrator to perform functions such as configure the configurator database and other system aspects, such as system security administration.

Another administrative application is the control panel application allowing an administrator, for example, to create, remove, start and stop services on local machines. Another function that may be provided by the control panel application for remote administration of services installed on the network.

A third client application that may serve administrative purposes is a tag activator which may be an application allowing the system administrator to activate and deactivate various tags. For example, when various tags are removed from operational use they should be deactivated to order to prevent unnecessarily populating client application with data. Alternatively, when tags are brought back online, the tag activator may be used to reactivate the various tags as associated with a particular asset or personnel.

Yet a fourth application which may be used for administrative purposes is the low battery viewer application. This application may allow the system administrator to monitor the battery status of the population of tags. For example, when a tag's battery is low, the low batter viewer may notify the system administrator to replace the battery.

In light of what was previously described, all services included in various embodiments of the server, for example in the system 2810 of Figure 27, incorporate both publisher and subscriber functions. These services, except for the broker services as previously described, may be located within the server 3056. Generally, the previously described publish/subscribe mechanism is also used whenever an application, such as client application 3052b, subscribes to data from a services. All client applications have a subscriber role, and none of the client applications are viewed as publishers in the previously described publish/subscribe model.

Configurator service. Generally, the configurator services may be termed as central services for the system. The services generally serve as a link between the configurator database and various services and applications. Whenever an application, such as a client application, or a service, such as one that may be included in the server, begins, it locates

the configurator service either by listening for a message from the configurator, or by begin told the configurator IP address and port by the user. The application then requests from the configurator start up information. The configurator service pulls information from the configurator database and forwards it to the service or application requesting it. The configurator application, as previously described as an administrative application, may be used for example to populate the configurator database with configuration information.

It should be noted that in one embodiment, the configurator IP address and Port ID may be found in the Windows NT Registry in which NT stores various system parameters. Applications may also store parameters here as well. In another embodiment, the configurator may store and broadcast this information as needed to services included in the system.

Broker service. Generally, when the services or an application wishes to subscribe to particular information, it calls or invokes the broker service. It tells the broker service what information it is looking for and the broker assesses its various publishers lists to see what publishers have registered for the particular information as requests. The broker service then refers the subscriber to the publisher. The broker then steps out of the transaction, and the subscribe and publisher then communicate directly with each other as previously described. Almost every service and application may use the broker services, every publisher registers with a broker. Generally a broker is visible on different subnets is used to allow services on the subnets to communicate.

Generally, an embodiment of the configurator service has two components. First there are a set of utilities for manipulating configuration information as stored in the

Alerter service. The alerter service subscribes to the location data converting into alert data using the user defined alert rule set. This rule set is created in the alert builder section of the configurator application. The rule format is described in more detail in paragraphs that follow. Once alert data has been generated, it is published to the alert services subscriber. In this particular embodiment, the recorder service is the only service that typically subscribes to the alerter services. Other applications may subscribe to the alerter service such as the alerter application which may display alert information to the user such as upon a display device.

Recorder service. The recorder service generally collects location and alert information form the location and alert services to which is subscribes. This data is then stored in the recorder database 2028 of Figure 27 which may be accessed by other software. Generally, in order to avoid having the recorder data store too much data or old data which may be considered stale in an particular instance, a policy should be adopted by the system in accordance with the needs with a particular application. For example, policy may be that data passes a certain age such as more than 7 days old, maybe deleted. Additionally, such policy may include a rule such that only the tags most recent 100 locations are stored in the recorder database.

In one embodiment, the recorder database may be used by applications that need historical data and information. While some applications may subscribe to real-time data, others may subscribe to historical and collected data over a time period. For example, the data included in the recorder database may be used by a tracker application for reports and audits of tag history. The tracker application may determine when a tag was last seen. The tracker application may also make data available in standard or predetermined formats.

It should be noted that in one embodiment of the system, a tracker application functioning as a client application may directly access the recorder database.

It should be noted that the various publish/subscribe connections established such as in the server established using TCP/IP protocol. Generally, the time required to make TCP/IP connections may vary with network load. Due to this variance, various components of the publish/subscribe client server controls may include a parameter, such as to connect timeout which sets the amount of time for an attempt a connection between a client or a subscriber and a server is timed out. A default value just for a second for the read/write property may also be set initially using the configurator and values contained in the configurator database.

It should be generally be noted that in a preferred embodiment there may be instances when this timeout value may be changed. For example, if a subscribe command is issued under a heavy network load, the 4 second default may be insufficient to make the proper network connection resulting in a failed attempt. Increasing the connect timeout variable value to a value greater than 4 will circumvent this problem. However, it should be noted that once this is raised, applications may be stalled as they wait for responses for longer time periods from publishers or subscribers that may have been terminated. It should be noted that once a connection has been established between a subscriber and a publisher, the publisher may dump data or transfer data to a subscriber without checking for confirmation that the data was received. Because the subscriber may have terminated without publishers knowledge, the publisher may periodically "ping" the client to ensure its continued operation. In order to remove nonexistent client subscription, if the periodic attempts fail for more than a certain period in accordance with the client timeout variable

database. Generally, these set of utilities interface with the database to manipulate the information contained therein. The second portion of the configurator includes services that publishes information or communicate it to other services for example within the server 3056. All other services subscribe to the configurator. It should be noted that in a preferred embodiment only the configurator directly interacts with the configurator database. Other services as may be included in the server 3056 obtain configurator data through the configurator service, as by use of the second portion of utilities previously described.

Tadder service. Generally, the tadder service consolidates and distributes TAD data. It should be noted that there may be multiple tadder services in one particular installation. In some instances, other tadder services may subscribe to a first tadder service. One particular example, a tadder service may collect TAD information from several other TAD publishers or cell controllers. One particular tadder service may serve as the central publisher for tadder information.

Locator service. The locator service subscribes to TAD data from tadder services and converts that data into location data using user defined location rule sets. Location rule sets are created in the location builder section of the configurator application. The format of location rules and data are described in more detail in paragraphs that follow. Once location data has been generated the location service publishes it to other subscribers. Services that may subscribe to the locator service include for example the alerter and recorder services as may be included in the server 3056. Similarly, other client applications may subscribe to the locator data such as a location viewer or a low battery application as previously described.

TAD data is the most fundamental type of data in the system. Generally, TAD data indicates the distance between the antenna and tag that responded to it signal. Software applications such as included in server 3056 convert the tag into formats named by the user. It may then be used to generate location data asked by the locator service within the server 3056.

TAD data in this particular embodiment may be generated by cell controllers such as 3050A through 3050C in Figure 57. Additionally, software applications may simulate cell controllers and also generate TAD data.

What is described in paragraphs that follow are the various data format such as TAD data, LOC data, ALR data, and TSD data as may be used and produced within the system.

Referring now to Figure 60 shown is an example of one embodiment of a format for TAD data format. In Figure 60, view 3090 shows the general format of a TAD message which includes, for example, the TAD indicator of "TAD" indicating a TAD message, and additionally tag specific information. For example, tag specific information may include the tag identifier, the number of antennas that have detected the tag, the cell controller which detected the tag, and the like. Field 3092 shows a specific instantiation of a TAD message as generally described in field 3090. Accordingly, field 3094 shows a breakdown or a parse of this TAD data indicating what each specific field means in accordance with the previously displayed data formats.

Referring now to Figure 61, shown is an example of an embodiment of a LOC message format. Field 3100 shows the general fields that are included in a LOC message.

values similar in function to the previously described connect timeout, the publisher may remove a subscriber from it subscription list. Generally in one preferred embodiment, the default value for client timeout is 3600 seconds or one hour.

The various services, administrative applications, client applications, and the publisher/subscribe mechanism generally exist to convert, transmit, organize and display the various data obtained and gathered by the system.

Referring now to Figure 58, shown is a data flow diagram in one embodiment of the invention illustrating the different data types with regard to TAD data. Generally, TAD data 3062 may be parsed 3072, or segregated out, to produce TSD tag data 3074. Generally, the parsed or TSD data is parsed, raw TAG data as gathered by the cell controller for its various content. Similarly, TAD data may be converted 3064 by the location service using a location rule set to produce LOC data 3066. Generally, LOC data is converted by the alerter service using the different rules such as the alert rule set 3068 to produce ALR data 3070.

Referring now to Figure 59, shown is an example of an embodiment of the various client application which may be included within the system and different types of data that they may use. For example, TAD data 3080 is used by the floor plan viewer and the TAD viewer. Generally, the floor plan viewer may display TAD data as associated with a particular floor or zone or subnet. Similarly, alert data 3082 is used as subscribed to the alert viewer 3088B which may display various types of alert data to a user in a readable format. Other types of data and the different types of applications which may use them are shown also in Figure 59.

Generally, the LOC message includes a LOC identifier and LOC is the initial portion of the message, followed by location specific information, such as the location in which the tag is currently located. Field 3102 shows a particular example of the generalized message format 3100. 3104 includes an interpretation of the information contained in the specific message 3102.

It should generally be noted that information previously described with the LOC and tag data formats may vary with a particular embodiment and should not be construed as limiting those principals described hereto related to the invention.

Referring now to Figure 62, shown is an example of an embodiment of the ALR or alert message format. Field 3106 shows the generic message format for an ALR message including an ALR indicator followed by various alert types of information encoded with a corresponding numeric value. Field 3108 shows an instantiation of what a message according to the format of 3106 may look like. Field 3110 describes that message as particularly set forth in 3108.

Referring now to Figure 63, shown is an example of a message format in accordance with the TSD data format. Generally, TSD messages or data formats are derived from tag data as obtained from the TAD message format. The TAD message format borrows information from the TAD message, and reformats it to adhere to the TSD format as set forth in Figure 63. Field 3112 shows the general format of the TSD message. Field 3114 shows an example of what a particular message following the format of field 3112 may look like. Field 3116 interprets the data as included in field 3114 in accordance with the format of field 3112.

Rule sets are generally used in this system to convert between data types. Generally, a location rule set is used in this embodiment converts TAD data into LOC data. Similarly, an alert rule set converts LOC data into ALR data in accordance with the previously described formats. Generally, a location rule set takes as an input tag data and tests it against a list of user defined statements or rules. Each rule is associated with a particular location. These rules are generally descriptions of location in terms of tag data. For example, a basic rule in pain English may read "a tag is in this location if it is within 10 feet of antenna 2 on the cell controller named CC1 and more than 20 feet from antenna 3 on the cell controller named CC1". Different types of rules may be defined using location rules sets. One type is called a "closest to" rule. A tag for example may be set to be in a location because it is less than 15 feet from a particular antenna. It can also be said to be in a location if it is closest to a particular antenna. Thus, a rule may be defined which states "a tag is in this location if it is closest to antenna 2 on the cell controller name CC1". Location rules may make use of logical operations. These rules may include operators such as logical operators AND, NOT, and OR, and the use of parenthetical expressions. Additionally, "closest to" rules may be used which define a rules in accordance with the proximity of a cell controller to an antenna, for example. All of these rules behave as may be expected in accordance with various logical and operator precedence.

Alert rule sets generally receive LOC data. Alert rule sets are used to convert this LOC data into ALR data format. The alert rule set as may be stored in the configurator database, is used to determine whether tags are in an alert condition. For instance, if tag one and tag two are in a particular room, and an alert rule has been defined and triggered, then an alert will be fired. An alert rule set may be defined by drag and drop as described in the configurator description which will follow. Generally, in this particular embodiment, there

may be 7 alert types four of which are single tag alert types and three of which are paired resource alert types. In other words, single resource alert types involved a single asset tag and the tracking system whereas other types of rules such as the pair resource alert types may involve two or more types of asset tags. This has been previously described.

It should be noted that in accordance with each particular embodiment, various network topologies have been setup and created such as those described in conjunction with embodiments such as Figure 52, it is necessary to load the various types of software onto subnets containing servers and cell controllers. For example, it may be necessary to load DHCP, IIS or name server software such as WINS onto the subnet that include servers and cell controller. Subsequently, server and client software may by loaded and configured.

In this particular embodiment, it is recommended that WINS, DHCP, and IIS services be setup on any subnet in which a server or cell controller group resides. Generally, these services may be configured prior to installing the server software. In this particular embodiment, cell controller should be on a sub net that has a DHCP server installed and setup. Generally, DHCP is used to assign static and IP addresses and hosts means to the cell controllers. This may be in contrast to the usual function to DHCP known to those skilled in the art as a provider of dynamic IP addresses. In this particular embodiment, this is done because the IP addresses and host name relationships are maintained in order consistently identify individual cell controllers.

Generally, there are two types of applications that may be used in this system. There are server applications, and there are client applications. In a preferred embodiment, it is recommended that a server be installed with the full compliment of services and application

as indicated in the forgoing description. Client applications on the other hand may need only be installed with those applications which may be used in accordance with the various uses of the system to monitor and track personnel and assets. Once the appropriate software is installed, it is necessary to configure the service software to address the needs of the site installation.

Two application which may be used to install software such as in a server 3056 are the configurator application and the control panel application previously described. Generally, the configurator application defines services, rule sets and tags in a particular installation. The control panel application may be used to install service software by actually activating services. The services then refer to the settings in the configurator application to set themselves up. Before starting configuration process, its important to have noted each of the systems in the network topology in accordance with the particular embodiment of the system and to have diagramed the configuration. It is also recommend in an embodiment that antenna layout and cell control layout completely define a set of location and alerts accordingly. The configurator program may be run a second time in order to change or modify any of the foregoing configuration settings. Generally, a configurator application is used to create data as stored in the configurator database defining the behavior of individual services. The control panel is an application used for remote administration of services. For example it may be used to create start and stop various services as included in the server 3056 of Figure 57 including the configurator service itself.

Regarding the configurator application, there are several techniques which one may use a particular implementation of the configurator application. The configurator may

include for example an express installation procedure which may be a default or rapid configuration path used with simple systems. Simple systems may include a least possible number of services as need for a system to function. For example, simple systems may be configured with no alerts. If additional information is need later on, the configurator primary interface may be used to alter this setup. Generally, a simple system may include basic elements as follows: One or more cell controllers, a single tadder, a single locator, a single alerter, if any, and a single recorder. A single locator for example which converts TAD data into LOC data, may automatically have a default set of location rule sets by defining locations in terms of closest antenna as defined in the configuration database. In other words, tagged location may be defined in location rules by whatever antenna it is closest to. This system may also include in a simple configuration or a default installation a single broker and as many tags as the administrator wishes to configure in accordance with the elements wide and should be tracked by the system. More powerful systems may also be configured that have multiple services such as multiple cell controllers, multiple tadders, multiple locators, multiple alerters, and multiple brokers. Each of these types of installation or configuration options is defined in accordance with the needs of a particular system.

In one embodiment, there maybe options for defining the configurator database 2814, of Figure 27 during system installation. For example, an existing database may be defined as a configurator database, or a completely new configurator database may be input and defined.

Generally, the configurator database contains detailed information about the setup of the system. Loading a preconfigured configurator database may allow circumvention of

the setup procedure and provide for configuring those elements that need configuring within a particular system.

One option as may be included in configurator application software, previously referred to as administrator software as an "express" installation procedure that may be a software option allowing the user to quickly configure a basic system. Alternatively, a standard installation procedure with less defaults than those set forth in the express installation procedure may allow the user to build a flexible and complex system that is needed in accordance with the various hardware and software system of a particular embodiment. Generally, in one particular embodiment, the standard installation procedure may include: setting up port settings, adding tags, creating cell controllers, setting various parameters of the cell controllers, creating brokers, creating tadders, creating locators, defining locator rule sets, creating rerouters, defining alerter rule sets, creating recorders, assigning brokers to publishers and subscribers, and defining cell controllers to subscribers, tadders to subscribers, locators to subscribers, and alerts to subscribers.

After an initial system and installation, such as via the previously described express or standard installation paths, a system may be configured again using a primary interface. Generally, this interface may be used to make modifications to the previous installation and in particular any particular component of the system. For example, in reconfiguring a portion of the system one may wish to reconfigure various port assignments, vary the number of cell controllers, or make modifications to various rules such as those included in the alerter service or the locator rule service. It should generally be noted for example that rules such as those used by the locator like the locator rules sets are stored in the configuration database and used, by various locator services.

In one particular embodiment, various tags may be grouped or associated together. One application such as the viewing application may have executed the client application such as 3052a or 3052b of Figure 57. An application such as a tag viewer is an example of one in which data is used as produced and propagated by the server system 3056. For example, a tag viewer application may display tags in a tree-like structure similar to a directory structure.

Referring now to Figure 64, shown is an example of a screen display as may be included in a tag viewer application. Shown in the left window 3120 is a list of groups in an expandable tree view. In this particular example, in window 3120 the root node is the active or current nod that is selected. This results for example in a display of all of the tags in the right most portion of the screen 3122. Alternatively, if one of the groups such as the patients or doctors group in the left portion 3120 of the screen are selected, only those tags in the current group are displayed in the right most portion 3122. Indicated on the screen as in Figure 64 in the right-hand portion of 3122 may be various types of tag information. This may include for example the serial number of the tag uniquely identifying the tag in the system. Additionally, resource name, resource type and an additional descriptions and the battery installation date may also be displayed. Generally, resource name is the name of the asset the tag is attached to. A naming scheme may be defined for various names to be determined by the various users of the system. The resource type may indicate the type of asset the tag is attached to for example, there may be two default asset types of personnel and assets. However, additional types may be created for example using different types of editors and application in the system. The description field also displayed may be a description of the resource to which the tag is attached. The battery installation date may also be displayed and relevant information regarding when the battery was placed on the

tag. Other information that may be included regarding the battery installed in a particular tag is the battery type, model number and the duty cycle. The battery type may indicate the documentation that came with the tag. The model number may be the model number of the tag such as the T20 or the T30. The duty cycle may indicate the frequency with which the tag sends out his signal. Once tags have been added to the system as an option for example they may be associated with various tag groups. Such a function may be also provided through the application in the system.

Additionally, an application may be used to add cell controllers as in the system. In one embodiment, the configurator may be used to add, remove, or delete cell controllers. Cell controllers may be added to the system once it is up and running. Various pieces of information associated with a particular cell controller may be displayed via an application. This information may include for example, a host name, a network interface card address and a IP address. Generally, the cell controller IP and host name may be set by an administrator using DHCP, and also recorded in the configurator database. Alternatively, another embodiment may store this data only in a single location.

The network interface card address in a cell controller is one that may be indicated on the cell controller itself. The IP address is that which is assigned to this particular cell controller in this system in the installation process. It should also be noted that the IP address is generally the same of that as which you find in DHCP software during the system installations. Generally an administrator may use the various applications such as the configurator application, to manage the operation of cell controllers changing parameters if necessary. It should be noted that modifying cell controller parameters is not generally recommended.

An application may exist in the system that may display various cell controller information. In one particular embodiment, there is a cell controller interface executing on the client system upon which the client application for example 3052B is executing. These types of information may be displayed using this application in this embodiment. For example, TAD data, antenna data, and antenna control information maybe collected and displayed. For example, antenna data may be display in the form of a graph for each of the antenna.

It should be noted that one or more brokers may be initially created. Additionally, once the system is started broker services may be added, removed or edited from the system. It should be noted that if only one broker is created which is the default, the subscribers and publishers may automatically be assigned to it significantly simplifying the installation procedure.

Generally, a tadder is set up by entering its host name, IP address, port ID and registration fields indicating what filter the publisher will be using. For example, the tadder named
ER may be publishing TAD1 data which is a subset of all of the TAD data, noted TAD* or TAG=*. Its registration field may be TAD1 indicating that it publishes data in accordance with matching subscription requests. A similar approach is used to set up locator, alerter and recorder type of services.

The location builder provides for building rule sets as used by the locator services. In this particular embodiment, as previously noted, its location rules may be stored in the configurator database. In this embodiment, it should be noted that the locator builder may

be included in the configurator. Generally, in discussing the creation of location rule sets, a hierarchically organization of locations is referred to as a tree view. On the tree view, there are nodes that will represent locations. Each node or location has its own location rule. There are several kinds of locations or nodes. These include for example, child locations, sibling locations, parent locations, and end locations. Child locations are generally those locations which are known hierarchally placed beneath another location known as the parent location. A similar analogy might be a subdirectory as the child with relation to a parent or root directory. Similar locations are those locations which share a parent and are on the same physical location level. Parent locations are those locations which have children locations. Parent locations can generally never have their own rule, but rather inherit the sum of the children rules. This generally means that any tags seen in a child location are also seen in its parent's location. End locations are the last level of locations in the tree view. These locations can have their own rules and by definition never parent locations.

Referring now to Figure 65, shown is an example of the location tree view showing relationships of various locations. Generally, in the location display 3130 a root includes in the first level of hierarchy including a base floor, a top floor and a second floor. Within the second floor hierarchy there are two elements, a janitorial room and a class room 22. The child location is a branch of a parent location in this particular example, the janitorial room is a child of the second floor. Locations may also be sibling locations. Sibling locations share a parent location in a particular example 3130 the base floor, top floor and second floor are all sibling locations. They all share the root location as the parent.

Referring now to Figure 66, shown is the location builder screen showing the list of hierarchically organized locations in one embodiment. The left frame 3140 contains a tree

or hierarchical or organized view of locations. The right portion of the screen contains the actual rules or rules by a selected location as well an expanded rule set and basic guidelines for creating new location rules. This shown in screen portion 3142. Location rules may take on different forms. In this particular embodiment, referring now to Figure 67 shown as the basic form for one of the rules as may be defined as a location rule. Field 3144 contains the generalized formal syntax of a location rule. For example, the cell controller name and antenna index are included. Additionally, a range is specified. In field 3146 an example of a rule in accordance with the general description in 3144 is defined. The rule 3146 means that any tag closer than 40 feet of an antenna 1 on a cell controller name CClobby is included on the node. Field 3148 shows a commonly used simple form as the closest to rule. It takes the form of CT(CClobby.one) which means that any tag which is closest to antenna 1 and cell controller named CClobby is included on the node. Similar to other statements, simple form rules as previously described may be combined into longer and more complex rule sets using different operators.

In this particular embodiment, four combination operators that may be used included to combine simple rules are: the | as an OR operator, & to form an AND argument, the NOT used to negate or create negative arguments, and the CT as previously described to create a closest to argument. Additionally, the use of parentheses expressions may allow for certain combinations of simple forms to be evaluated in accordance with the precedence are defined by the user. It should be noted that the use of parentheses in this particular embodiment provides for combination and evaluation of operators and expressions in accordance with those rules of standard logic.

With regard with location rule sets, locations are evaluated by order of preference. Incoming tags are placed only in a single location. For that reason it may be necessary not only to create locations, but also to prioritize them. Locations higher in the tree view are evaluated before locations lower in the tree view. There may be various ways in an embodiment to prioritize locations. For example, one may select, as by right clicking a mouse button, on the location to be reprioritize. Also a location may be prioritized by accessing the pull down menu on an icon that may be displayed on a menu of a user interface.

Other facilities may be included in a particular embodiment of a location builder. For example, a macro facility may be included in the location builder to facilitate the task of entering location rules. In one particular embodiment, the macro facility mimics that of the Microsoft Word macro facility automatically replacing one string with a combination of longer strings. Thus, a two character combination, for example, may be replaced by a complex statement used in building a rule.

Generally, the locator service which uses these rules uses techniques to prevent stationary tags from appearing to move or jumping among nearby locations. When location rules define contiguous zones, variations in TAD data may cause stationary tags to appear to incorrectly jump back and forth between zones. Therefore, the locator service may include techniques to prevent stationary tags from seeming to "jump" between nearby locations. In one embodiment, if a tag is already in a location, a check may be performed to ensure that a new TAD reading is consistent with that location. If it is, other location rules are not checked and the tag is treated as being stationary. This, in conjunction with defining overlapping locations, may eliminate tag jumping. Additionally, when new data is received

that is different from previously received data, locations are not modified until that new data is received twice in a row. Higher reliability may be achieved by requiring that the same data be received more than two consecutive times. This technique may cover situations, for example, when specific antennas temporarily are not in use for various reasons, or when distance measures may vary due to multipath effects.

The following describes alert message types in building alert rules accordingly. Generally, in this particular embodiment, there are seven predefined alert message types referred herein to as single resource alert types, and several types which are paired resource types. All alerts have a delay property which indicates the amount of time which should pass before an alert is triggered. This reason may prevent, for example, individual or single readings from setting off an alert which are not really indicative of an alert situation. For example, only alerts that are confirmed for 5 seconds will be fired. Single alert resource types may include, for example, entry alerts, exit alerts, stationary alerts and timeout alerts. Generally, an entry alert is triggered when a resource enters a location. Similarly, an exit alert is triggered when a resource exits a particular location. The stationary alert is triggered when an associated resource is in an indicated location for a specified period of time. For example, this might indicate when a resource, such as a wheelchair, is seeing little use. The timeout alert may be fired, for example, when a resource is last heard from in an indicated location but has not been heard from in any location for this set period of time. Subsequent chirps from this resource in the location clears this alert. For example, this alert of a timeout may be triggered if a resource is removed via a second story window without entering another location. The duration attribute sets the amount of time that the resource has not been heard from.

Pair resource alert types include an escort, a friend, and a foe alert in this embodiment. The escort alert is triggered when the resource selected first in the pair enters a particular location without the second paired resource. The friend alert are two resources that are supposed to be together are not in the same location. The foe alter is triggered when two resources that are supposed to be separated are in the same location.

Referring now to Figure 68, shown is an example of an embodiment of the alert builder screen. Referring now to Figure 68 the alert builder screen is shown as being divided into three regions 3150, 3152 and 3154. The first region 3150 features a list of resources currently stored by the configuration. The second region 3152 shows the location rule set upon which the alerter is based. The third region 3154 displays the actual alerts defined in the alert rule set. An example of how this area is used to add a single resource alert, a resource is dragged from the list in area 3150 onto a location in the area 3152. After filling in a dialog box of subsequently the alert would appear in the area 3154. Editing and removal of alerts occurs in the area 3154. Additionally, referring now to Figure 69, shown is a defined alert table which may be used to show two fields indicating whether resource one and resource two as displayed in accordance with field 3154 of Figure 68 are tags or groups of tags. For example, as indicated in 3160 a true value means they are tags and false value they are tag groups.

In this particular embodiment, different types of multi-media data files, such as sound files, may be associated with a particular alert. For example, this embodiment a sound file my be entered into a dialog box to indicate a particular sound file to be played when an alert condition has occurred. Subsequently, this particular file should be copied to a predefined directory in accordance with where the alerter service will go in the occurrence

of an alert condition to find the file. Generally, it should be noted that the default types of directories are the types of information known to one skilled in the art that is defined with the configuration application.

A standard installation may have just one broker. It should be noted however that other embodiments of the invention may use no brokers, or have multiple brokers and accordingly have brokers registering with each other. Thus, if there are multiple brokers and a broker receives a subscription request, the response may include a redirection to another broker. Additionally, another embodiment of the system may provide information for multiple broker with a single subscription request. This may be appropriate in two particular instances. For example, if a broker can not provide the requested data, it is reasonable to see if other brokers may also provide this data. Secondly, if the system is heavily burdened, it is appropriate for the broker to search for other ways to get the same data providing a form of load balancing.

As previously described, the control panel application is an administrative application allowing an administrator for example to create, remove, start and stop services on local machines. Additionally, the control panel may allow for the remote administration of services. This application may include for example the ability to view publishers, view subscribers, reboot cell controllers, retrieve cell controller configuration information, clear locations, and reset individual alerts. In this particular embodiment, the configuration service must be started before any other services because the configuration provides start up information for every type of service. The control panel may be executed after the configuration application has been run on the central server. The control panel is executed on each system and is designated as a host of the service. Generally, the first system of

which the control panel should be executed is the system with the configuration database. To create and start services on that system, installer must first ensure that the configurators host name, IP address and port ID are correctly identified when the services to be run are selected. By default, the configuration, tadder, locator, alerter, recorder, broker, and tagged data services are selected. By default, all except the recorder are set to start automatically when Windows and the operating system starts for example. Additionally, an administrator may stop and start individual services. Services run according to the properties set up in the configuration.

It should be noted that a user must generally be logged in as an administrator on a remote system to obtain certain functions or the ability to do certain things within the system for security reasons. For example, system administrator should be logged on as an administrator of a domain or have certain administrators rights. For example in order to start and stop certain system services or have accesses to different types of data in the databases.

The previously described low battery application is one type of application than an administrator may use to monitor the battery status of various tags. The low battery application may receive low battery notices from the tag data service which in turn derives its data from the tadder service. Recall previously as described that the TDS transforms information from a tag message into a TDS format as needed by applications such as the low battery application.

As previously described various types of client applications such as 3052a and 3052b in Figure 57 may require configuration. These include for example applications

which provide for viewing certain types of information in the system. One particular application is known as the floor plan viewer to serve as a unified viewer of TAD, LOC, and the alert data. The application functions by subscribing to data from various services such as the alerter, tadder and location services and superimposing data on a floor plan. In order for this outfit to work a floor plan in a predetermined format, such as AutoCAD, is generally required. A tool for viewing such as WHIP! may be used to display a floor plan. This is read for example by the floor plan via software in conjunction with the data and displayed accordingly.

Similarly, the location viewer application may be used to show users what resources are in a given location as done by clicking on a particular floor or location within the tree.

Referring now to Figure 70, shown is an example of a user interface display to be used for a location viewer to show or display what resources are in any given location. In this particular embodiment, this is done by presenting a location tree as shown in the screen 3170 when a user clicks on the location such as the display on the left-hand portion of the screen 3170a, the tags in that location are display in the right-hand portion of the application window 3170b. This parent locations are the sum of their child locations. In this particular embodiment, clicking or selecting a parent location displays tags located in each of the child locations as well. Accordingly, clicking on the root location displays every tag in a particular location. In this particular embodiment, the location viewer is a dynamic application. Location data will automatically be updated as it is changed in the system in accordance with the movement of assets or tags associated therewith.

In a particular embodiment, by default, the location viewer application may display data for every tag in the system. However, the user may wish to monitor only a selected group of tags. A function in the software may be included for example to filter tags that a location viewer displays. This may be done by using groups or by identifying a particular tag ID to be monitored or removed from the viewing process.

What is described in paragraphs that follow are steps in one embodiment for creating applications as may be used in the system of Figure 27 and in the system of Figure 57 for example. Generally, it will be described in paragraphs that follow are a publisher and subscriber application. Generally, the publisher/subscriber system previously described uses the TCP/IP protocol to pass data between various elements of an application. The TCP/IP protocol functions in this embodiment are embedded in the client and server as ActiveX controls. The client/server controls may in turn be embedded in the published and subscribed controls. Finally, more advanced controls such as locator, broker, and alerter controls use publish and subscribe controls as their means of communication as well.

What will be described first are the fundamental publishing, properties, methods and events in one embodiment. This includes, for example, a *start publish method*, a *published message* method, a *host IP* property and an *error event*. Additionally, in our subscriber example with regard to particular methods, properties, and events, the *subscribe direct* method, the *unsubscribe direct* method, the data and event and the error event will be described.

It should be noted that the steps which will be described in paragraph set follow refer to one embodiment for creating a publisher application using the previously described

principles described herein. In this particular example, Visual Basic is used on a particular system to develop a publisher application. Thus, the first step in this particular embodiment is to launch the Visual Basic application. Once this is done in this particular embodiment, a screen with the user end interface will be displayed.

Referring now to Figure 71, shown as a sample opening screen as may be displayed when a user builds their publishing application using Microsoft Visual Basic. From the menu displayed in Figure 71, from the user end interface screen 3180, the standard EXE on 3182 is selected as with the mouse button. This causes the display of a blank project form. The next step in this process using Visual Basic is to load the particular publish controls in the tool box used by Visual Basic. In this particular embodiment, for example, this may be done by right-clicking on the tool box that may be displayed by Visual Basic selecting components and the control tab.

Generally, the tool box with the various publisher control objects are defined and used in defining objects that will be used by a publisher. Using the previously described publisher control tool kit with Visual Basic, a form is created which defines objects included in the form. In this particular embodiment with Visual Basic, an object oriented programming is used to define objects which will be used by the publisher. These objects, as in any object oriented programming technique, may be operated upon by various associated methods. This type of object-oriented programming is facilitated through the use of software such as the Visual Basic tool box with publisher control. What will now be described are the various objects that are created and included in a form using this particular tool box. For example, various text or label combination boxes and buttons are created and are associated with various objects.

One embodiment of a resulting form is shown in Figure 72. The form 3190 includes a publisher box 3192 included in this form 3192 are various fields which are objects as created with the tool box. In particular for this publisher is a text box label combination 3194 with a port ID field of the publisher. Also included in form 3192 is a start publisher button 3194a which will start the publisher online as the publisher of data when associated with code. The field 3196 indicates a string which is to be published as data by this publisher when brought online by clicking on the start publisher button 3194a. The button 3196a is used to actually publish the data specified in the field 3196. Recall that this publishing of the data, as by clicking on the button 3196a, makes available where publishes to all subscribers the data in the field 3196. Also included in form 3192 is an IP address field 3198a and an error field 3198b. The IP address field 3198a is the IP address of the publisher. The error 3198b field displays errors returned by the publisher. Using the tool kit, once this form is created with the objects basically all of the objects having been defined, no code is associated with each of the buttons. Thus, at this point in the process, the form has been created by defining the objects. However, no methods have been associated with them and thus no code has been associated with objects on the form. Thus, the objects will not perform any function in an application.

The next step in the process using Visual Basic with the object oriented programming technique is to make the form 3192 functional by adding code to the various objects created.

Referring now to Figure 73, code may be created and associated with the various objects as created. Referring now to Figure 73, code is added to and associated with the publish button 3196A which is responsible for publishing the data in the publish string field

3196. The field 3200 shows a short code snipit that will be executed when and associated with the clicking or selecting of the publish button 3196a. As an explanation of the code which is done in Visual Basic interfacing with the different methods as included in the API 3054 of Figure 57, the line 3200b simply creates storage used to capture the response from the publish message method of the system as indicated line 3200c. Line 3200c includes a routine call to the method from the publish control tool box. In this particular embodiment, two parameters are passed to this routine which includes the text as the first parameter to be published, and a second string which identifies the kind of data being passed. In this particular information, it is identified as the tad data type previously described which indicates to the publish message which subscribers to send the data to. Once this code has been added and associated with the publish button 3196a, when the publish button is pressed or selected, the text of the publish string is passed to the publish message method of the system. Similarly, code is created and associated with the start publish button 3194A. This routine includes a call to the *start publish* method registering or initializing the publisher with the port ID as specified in field 3194 of the form 3192. Similarly, a method is used to publish the host IP address as specified in field 3198A. A call is also made to the method for error event which causes any error event that is triggered to have the error box filled in with information about that error which has just occurred.

The publisher interacts with a subscriber. What will now be described is the development of a subscriber form with objects and the method or code associated with the various objects created in the form.

Referring now to Figure 74, shown is a form created displaying objects of a subscriber application. The form 3210 includes an IP field 3210a, and a port ID field

3210b. The form also includes a subscribe button 3210c, a message from the publisher text field 3210b and an error field 3210e. Generally, the subscribe button 3210 subscribes to the publisher application using the data as identified and associated with clicking on the subscribe button. For example, the code created and associated with the subscribe button 3210c includes a call to a *subscribe* method which includes several parameters. In particular, one of the parameters identifies the type of data such as TAD data for which the subscriber is to receive from a publisher. Alternatively, the subscriber may subscribe to LOC data or some other type of data provided by the services of the server 3056. The TAD data may also be subsetted out in accordance with various subnets or other areas of a location being monitored. For example, there may be TAD1 and TAD2 subset data which make up all of the TAD data. This type of data to which the subscriber subscribes is indicated as a parameter in the code such that when the subscribe button 3210c is selected, this subscriber is added to the subscription list and the other method steps previously described in accordance with the flowcharts and figures. Also passed as a parameter in the *subscribe* call is the address and port identifying the publisher which the subscriber which to subscribe to.

Similar to that which has previously been described, various methods are associated with various objects or actions and application may take. For example, in this embodiment, it is recommend that there is an *unsubscribe* mechanism or call in place for when the application closes. This would contain for example an unsubscribe call for each of the previously executed subscribed commands. Similarly, code is written and a call is made to the appropriate message such as a data and method to display message or publish data as received by the subscriber in the field 3210d.

After the code is written for the subscriber and the publisher as previously described, the two applications are complete. The code associated with this as known to those skilled in the art is created by Visual Basic and may be stored as projects. The next step in the process is to actually test the applications created and associated with the forms for the subscriber application and the publisher application just created. Subsequently, if the machine executable code for the publisher and the subscriber are executed, a communication is seen between the two in terms of the publisher's published data appearing on the subscriber's interface. For example, using the previously created forms and calls associated with them and executing the applications created with Visual Basic, the publisher start publisher button, the subscriber's subscribe button, and the publisher's publish button are pressed. The message from the publisher's publish string box appears as the subscriber's message from the publisher box.

An example of this is shown in Figure 75. The publisher application is displayed on the left-hand portion 3220 and the subscriber application is displayed on the right-hand portion as 3222. By clicking on the publish button in the window 3220, the string which appears in the 3220a publish string field is published and appears in the subscribers window 3222a. This illustrates an example of an embodiment where the publisher publishes the string "hello world," and the subscriber accordingly receives this data. Similarly, using the previous publisher subscriber model, the publisher may publish to multiple subscribers at multiple instances of subscriber application are created and launched.

Previously described is an example of how a publisher and a subscriber may be created using Visual Basic. It should be noted that other embodiments may be used to create subscriber and publisher machine executable programs or processes. Additionally,

the previous example shows the communication of a string. In this particular embodiment regarding object tracking and location, the fundamental type of message passed by a publisher which is a cell controller is the TAD message. The TAD message passes tag antenna distance information between a publisher and subscriber.

In this particular embodiment, the software of this system uses data generated when a cell controller passes TAD information to its subscribers. In one particular embodiment, one of the subscribers previously mentioned is a TAD parser which parses the TAD information to display, for example, information in a different format to a user. In one embodiment, routines may be added to extract particular values relevant to a format specified in the TAD data format as previously described. For example, routines may be written to extract parameter 1, parameter 5 and the like and return the result to the subscriber. It should be noted that in this particular embodiment, in order to indicate via subscriber or a publisher all of the tad data, notations such as "TAD *", or "TAG=*" may be used. To indicate a portion of that data, for example, "TAD1", or "TAG=1" may be used as associated with a particular system configuration. As previously described, the locator service converts TAD data to LOC data using a pre-determined table.

A subscriber application may also contain code enabling subscription to particular ALR messages. For example, when a specific alert condition occurs in the system, the subscriber may be notified of the occurrence of this condition by the alerter service. Function is provided in the tool kit with various methods for accessing the different types and fields as parsed in the alert data format. This is similar to that as previously described for TAD and LOC data which needs to be accessed by an application or subscriber.

It should be noted that the methods included in the foregoing description in this particular embodiment may be implemented, for example, as a library of routines available to a user. In one particular embodiment, these routines are included in a library which exists in a Microsoft system known as a COM (Component Object Model) dynamic link library ("DLL").

Referring now to Figure 76, shown is an example of an embodiment of a representation of the interface capabilities provided to a subscriber or client application. Generally, shown in Figure 76 are functions that may be implemented as API calls using ActiveX by Microsoft which is one implementation embodying object oriented programming techniques. Other embodiments may provide interface functions as may be used by a client application in other forms in accordance with a particular embodiment.

Included in Figure 76 are properties 3330, methods 3332, and events 3334. Generally, properties are attributes that describe the subscriber. A property may be set or modified and read in one embodiment through various APIs as indicated in Figure 76. For example, an API may be included to set and read the PORTID as indicated by 3330a.

Methods 3332 operate with regard to establishing or disconnecting a communication channel in this embodiment. For example, a subscriber may establish or subscribe to various types of data using either a *subscribe 3332a* or *subscribe direct* 3332b API. The direct subscription establishes a communication between a particular publisher and a subscriber. In contrast, the subscribe API established a communication channel between the broker and the subscriber to facilitate further connection between a subscriber and a publisher as previously described. A communication channel may be disconnected using

the *unsubscribe* 3332c or *unsubscribe direct* 3332d API. It should be noted that a publisher may also subscribe in manner similar in which an application may be both a client and a server. For example, if a publisher of data further forward this to a second publisher, this second publisher is also a subscriber of the first publisher's data and, accordingly, may also use the proper subscribe API's for communication. Subscribedirect may be used, for example, when the IP address of a data publisher is known. Subscribe may be used when the broker is used to direct a request and figures out the IP address of a publisher. The end results are that the subscriber obtains data in both instances.

Events 3334 are generally the result of an asynchronous event being triggerred. In this embodiment, APIs allowing an application to read properties may be used to obtain various event data.

Referring now to Figure 77, shown is an example of an embodiment of a representation of the interface capabilities provided to a publisher application. Figure 77, similar to Figure 76, includes properties 3340, methods 3342, and events 3344. However, these functions provided are with respect to APIs as may be used by a publisher.

It should be noted that various embodiments may include various components and services in the server. For example, if the IP address of a cell controller is known, one can directly subscribe to TAD information. Thus, tadder may not be required. Tadder may be a service that consolidates information from multiple cell controllers, but the information may be obtained using the direct subscription technique as well.

An embodiment may not include a broker. Similarly, the broker may be bypassed if subscribedirect is always used.

One embodiment may use the configurator to eliminate the need to configure each application individually. However, another embodiment may provide for individual configuration rather than using a central configurator.

Another embodiment may not use a location service if locations are mapped directly to readers.

The foregoing describes a technique for collecting data from a network of readers as provided by PinPoint 3D-iD systems, processing that data through a variety of services, and then distributing that data to applications and other services. People skilled in the art will note that the general techniques and the particular embodiments described herein may be applied to a wide variety of data collection devices in general, and identification devices.

For example., passive RFID readers collect information from tags in close proximity to their antennas. The same TAD message may be used to report information from those readers. While such readers may or may not measure distance to tags, they have limited read range and distance and may reasonably be reported as a fixed number, such as two feet. Such readers may be installed to detect passage of a tag through a portal, and a "closest to" location rule may provide a good indication of the tag's actual location.

Identification devices do not have to be radio based to make use of the publish/subscribe techniques and embodiments described herein. Bar code readers, infrared

identification systems (IRID systems), fingerprint readers, card readers and the like may report identification information through the same infrastructure. Any reader that is installed as a network appliance faces similar system-level software challenges that may be solved by implementing data publisher functionality in the reader. Applications may also be developed that are independent of identification technology chosen.

As is well-known to those skilled in the art, 3D-iD and similar tags may be integrated with other devices, such as temperature sensors or medical infusion pumps, providing a low-bandwidth wireless data line to those devices. Thus, the tags may be used as inexpensive radios in addition to supplying device location. Such information may be distributed through the network using the publish/subscribe technique and embodiment as described herein.

The publish/subscribe technique and the embodiments disclosed herein may be easily extended to support a variety of data collection devices. For example, medical telemetry systems wirelessly report medical diagnostic data, such as EKG information, to a nurse's workstation and other devices on the network. The publish/subscribe technique disclosed herein provides a clean and logical means to filter and move data through the network, using different message types and host services. Such an implementation is particularly attractive where there is also a need to collect and distribute identification information as disclosed herein.

The foregoing focuses on collecting information from readers which act as publishers, but the reverse is also possible depending on the application. In other words, the host and/or configurator may act as publishers to provide data and/or configuration

information to the readers which may act as subscribers. The readers in turn may pass such information to tags that are read/write enabled.

The publish/subscribe technique described herein is an efficient and practical way to distribute asset tag information across a network and is summarized in Figure 26A. Additionally, some users may want to store and access tag information using industry-standard enterprise management tools, such as Oracle, Microsoft SQL Server, and Sybase. To support this, Figure 26B shows a version of a recorder that may archive information to an SQL compliant database utilizing ODBC standard for database access. Additionally, referring now to Figure 26C, shown is an ODBC data feed connecting to a third party Warehouse Management System (WMS) or other vertical market application, for example. All that is needed is to add the appropriate schema to the WMS, and the standard ODBC data feed may populate the third party database accordingly.

Data to be saved via an ODBC-recorder may depend on user requirements. In one flexible and simple embodiment, a full archive of tag location changes, TSD changes, and tag alerts may be archived, with TSD changes and alerts triggering stored procedures in the database as per user requirements. This technique provides live information in the database and detailed historical information. This may potentially create a large archive of information and it may be best to therefore periodically run a program, such as weekly, to extract information of long term interest for audit and analysis. Such archiving software may vary in accordance with system and user requirements. In an alternative embodiment, the Recorder may subscribe to raw data directly from cell controllers and other services in archive them, for example, for debugging purposes.

Having described preferred embodiments of the invention, it will now become apparent to those of skill in the art that other embodiments incorporating its concepts may be provided. It is felt, therefore, that this invention should not be limited to the disclosed embodiments, but rather should be limited only by the spirit and scope of the appended claims.

What is Claimed is:

1. A location system comprising:

cell controllers for gathering data as transmitted by a tag;

a server for receiving and processing said data making it available for use by a client; and

a client that requests said data from said server.

2. The system of Claim 1, wherein said tag is associated with an object.

3. The system of Claim 2, wherein said tag is associated with a person.

4. The system of Claim 1, wherein said client is a client application.

5. The system of Claim 1, wherein said client is a database.

6. The system of Claim 1, wherein said server includes:

a configurator for managing configuration data;

a tadder for receiving said data from said cell controllers;

a locator for receiving and transforming said data in accordance with predefined rules; and

a broker for establishing a communication channel between a subscriber and a publisher of data.

7. A method executed in a computer system for publishing data in a location system to a subscriber comprising:

a data publisher registering with a broker;

the broker adding the data publisher to a list;

a subscriber sending a subscription request to the broker;

the broker examining the list to determine if there is a publisher to provide the subscription request;

the broker sending a message to the subscriber, the message including a network location of the data publisher; and

the subscriber directly connecting to the data publisher at the network location.

8. The method of Claim 7, further including:

the data publisher sending a return message to the subscriber indicating acceptance of the subscription.

9. The method of Claim 7, wherein the data publisher and the subscriber communicate using a predefined interface.

10. The method of Claim 7, wherein said subscription requests includes a request for data produced by a tag in a location determination system.

11. A method executed in a computer system for publishing data in a location system to a subscriber comprising:

a subscriber sending a subscription request to the broker;

determining, by said broker, if there is a publisher to provide the subscription request;

the broker adding the subscriber to a pending list upon determining that there is no publisher to provide the subscription request;

a publisher registering with the broker;

the broker determining that the publisher publishes data of the subscription request;

the broker sending a message to the subscriber including a network address of the publisher; and

the subscriber establishing a communication channel with the publisher at the network address.

12. The method of Claim 11, further including:

the publisher sending an acceptance message to the subscriber indicating acceptance of the subscription.

13. The method of Claim 11, wherein said subscription requests includes a request for data produced by a tag in a location determination system.

14. The method of Claim 11 wherein, communication between the subscriber, the publisher and the broker use a predefined interface.

15. A location determination system including:

a server that includes a first computer processor and software for configuring the location determination system and gathering distance data from at least one cell controller;

the cell controller connected to at least one antenna;

at least one client that includes a second computer processor and software for receiving and processing the distance data; and

a communication medium connected to the cell controller, the server and the at least one clients.

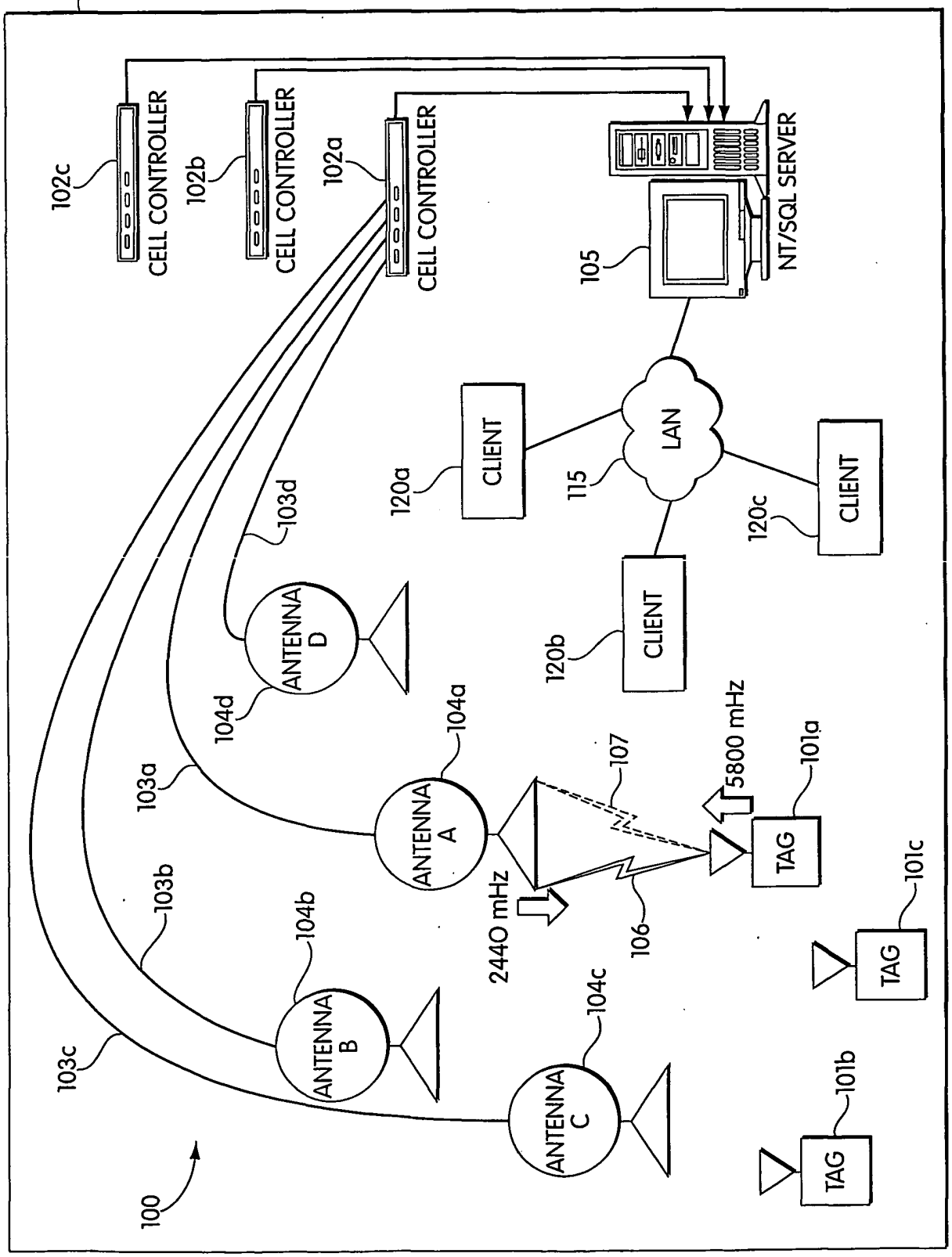

110
CELL CONTROLLER
102c
CELL CONTROLLER
102b
CELL CONTROLLER
102a
NT/SQL SERVER
105
CLIENT
120a
LAN
115
CLIENT
120c
CLIENT
120b
103d
ANTENNA D
104d
104a
ANTENNA A
103a
107
5800 mHz
101a
TAG
2440 mHz
106
103b
104b
ANTENNA B
104c
ANTENNA C
103c
TAG
101c
TAG
101b
100

Fig. 1

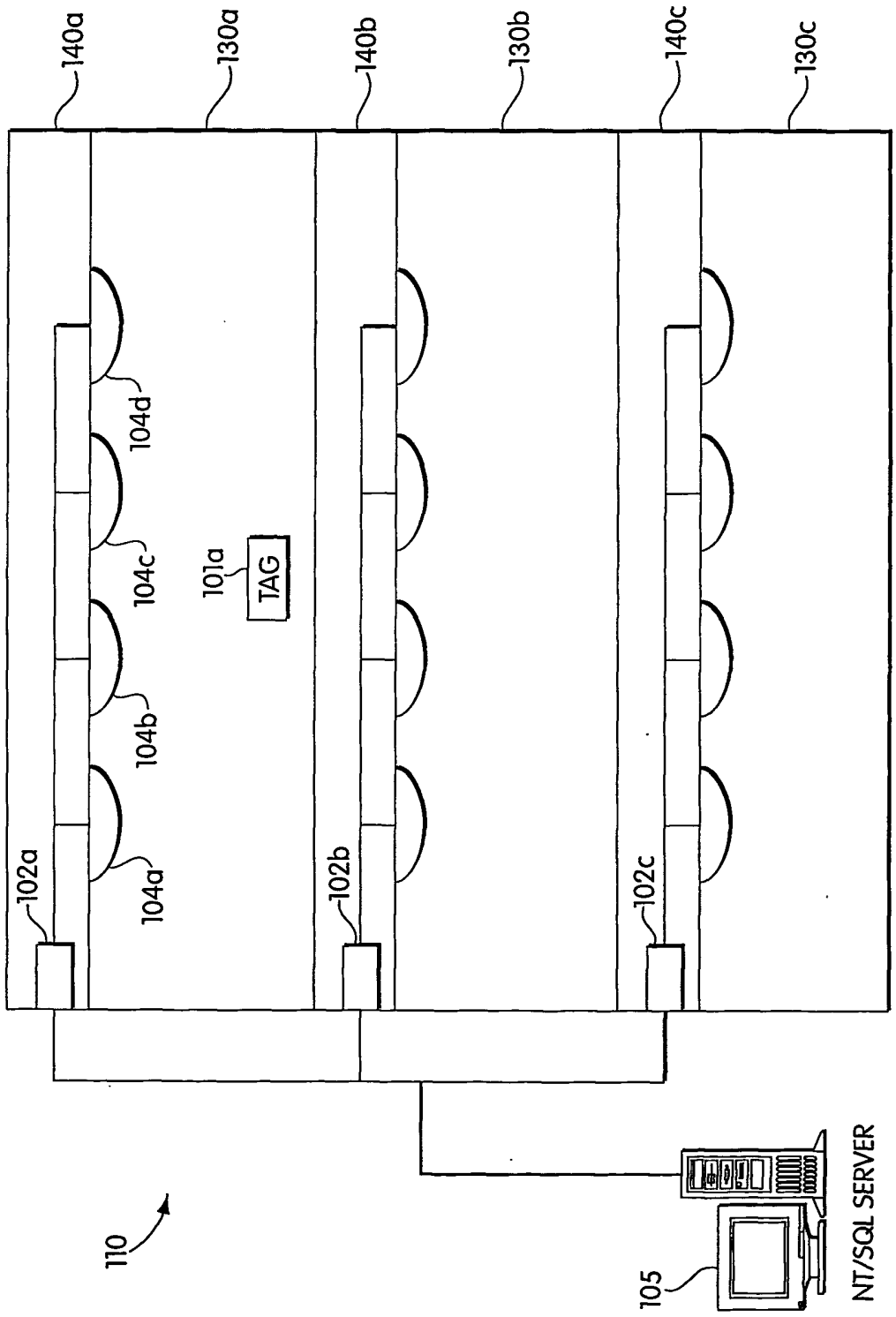

140a
130a
140b
130b
140c
130c
104d
104c
104b
104a
101a
TAG
102a
102b
102c
110
105
NT/SQL SERVER

Fig. 2

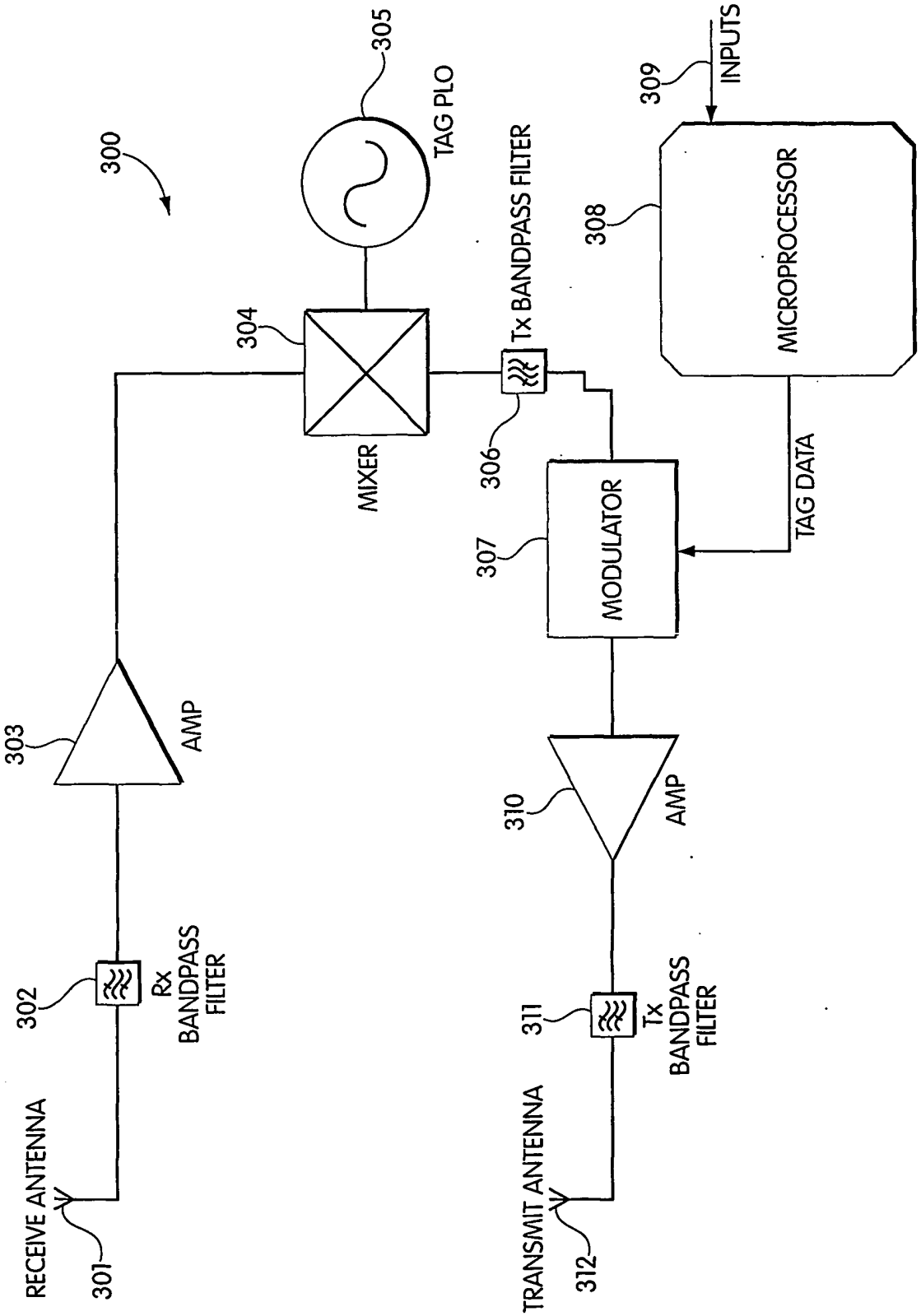

300
305
TAG PLO
309
INPUTS
308
MICROPROCESSOR
304
Tx BANDPASS FILTER
MIXER
306
TAG DATA
307
MODULATOR
303
AMP
310
AMP
302
Rx BANDPASS FILTER
311
Tx BANDPASS FILTER
RECEIVE ANTENNA
301
TRANSMIT ANTENNA
312

Fig. 3

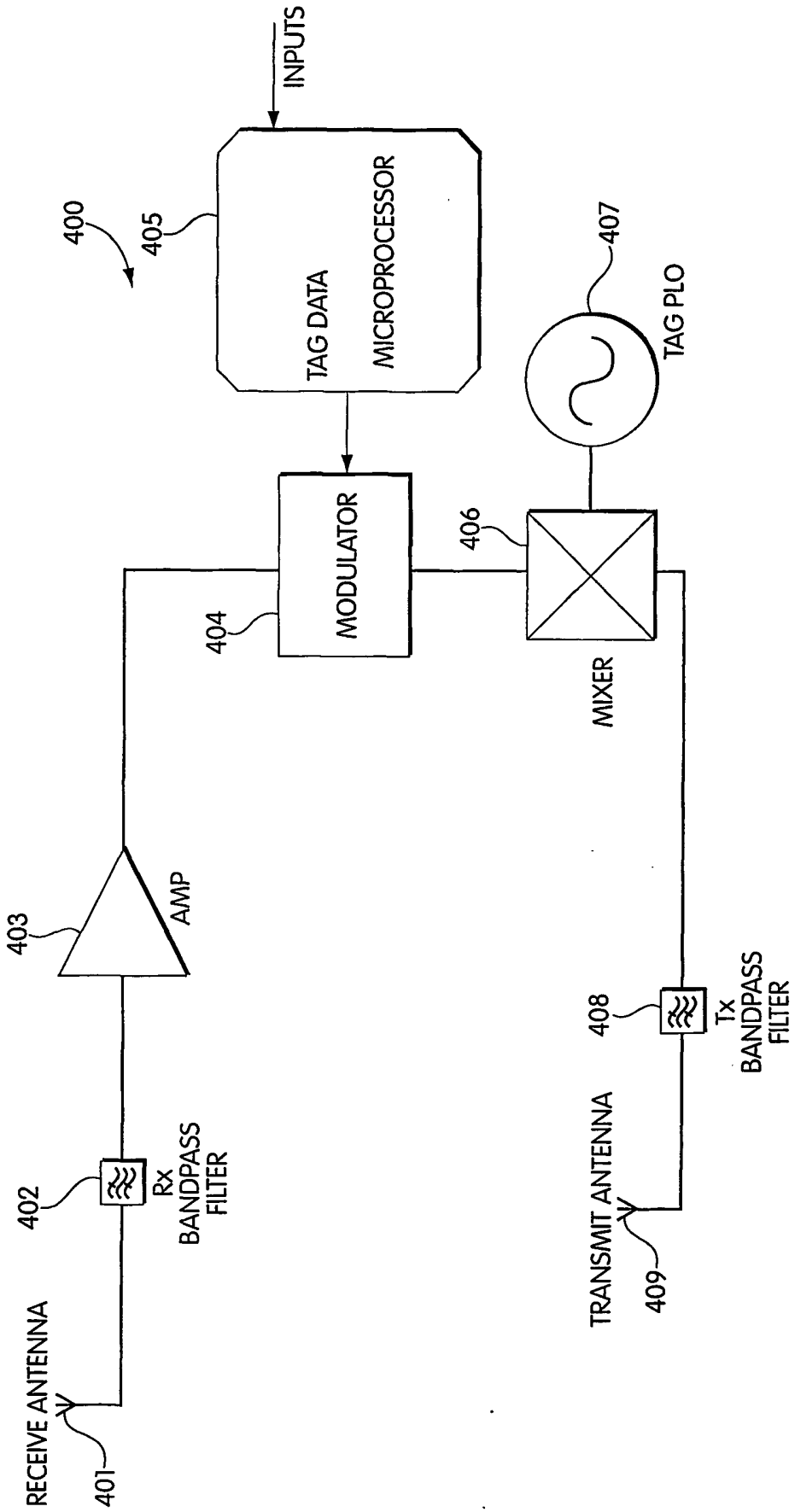

INPUTS
400
405
TAG DATA MICROPROCESSOR
407
TAG PLO
MODULATOR
404
406
MIXER
AMP
403
408
Tx BANDPASS FILTER
402
Rx BANDPASS FILTER
TRANSMIT ANTENNA
409
RECEIVE ANTENNA
401

Fig. 4

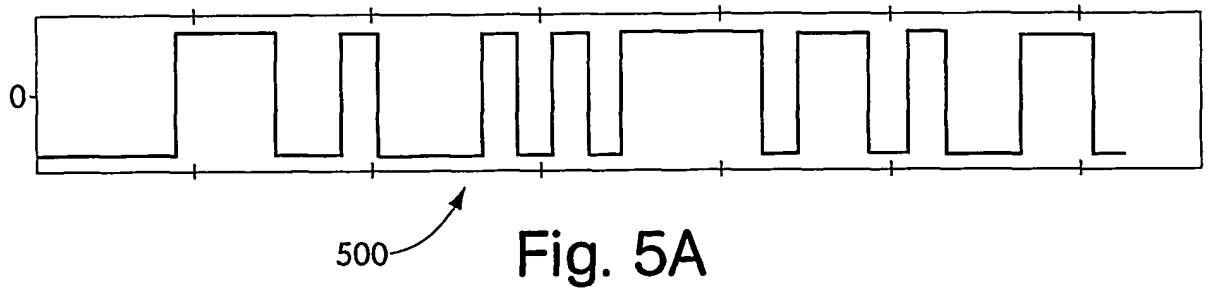



Fig. 5A

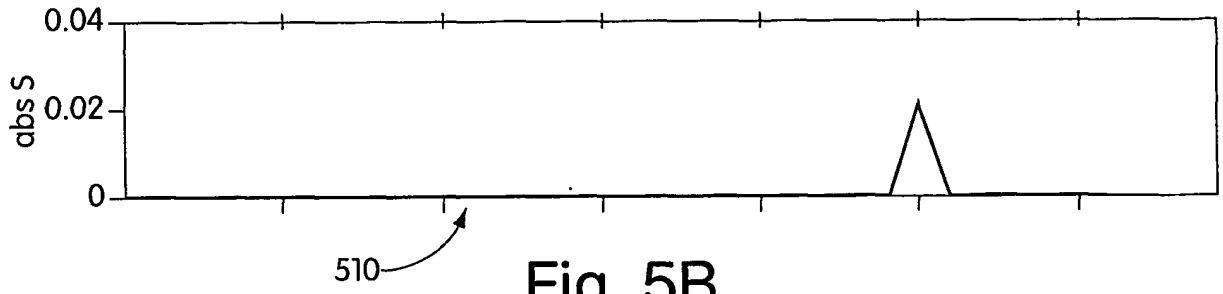



Fig. 5B

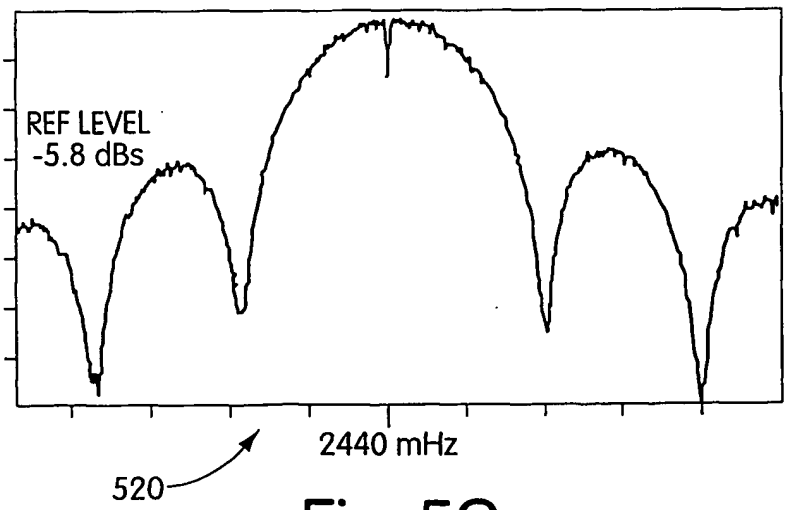



Fig. 5C

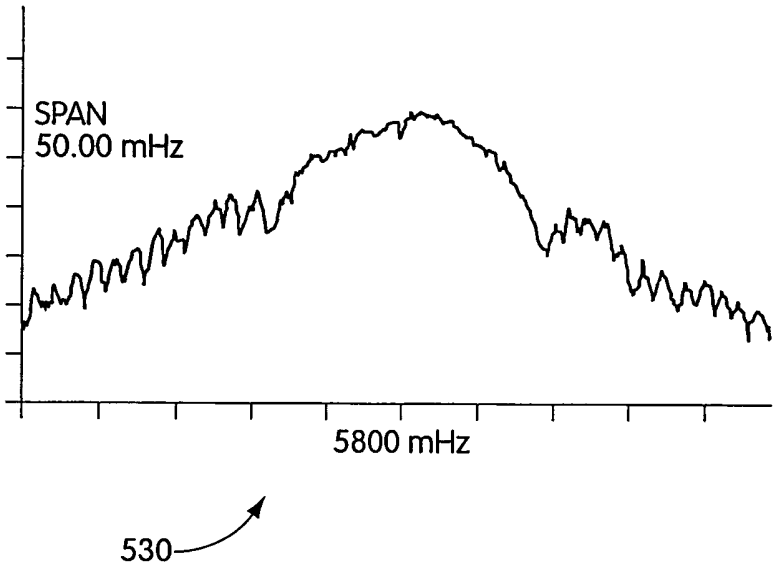

SPAN
50.00 mHz
5800 mHz
530

Fig. 5D

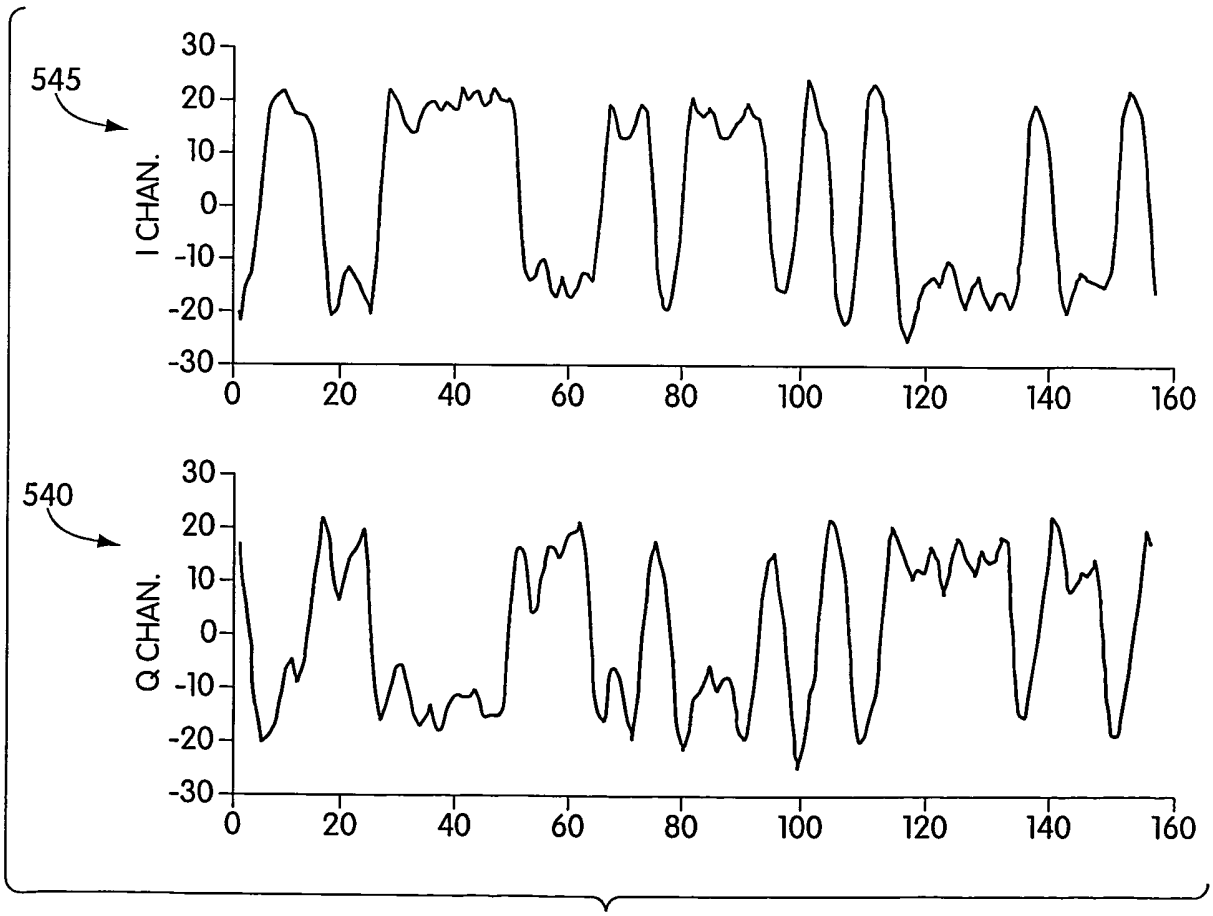

545
I CHAN.
30
20
10
0
-10
-20
-30
0
20
40
60
80
100
120
140
160
540
Q CHAN.
30
20
10
0
-10
-20
-30
0
20
40
60
80
100
120
140
160

Fig. 5E

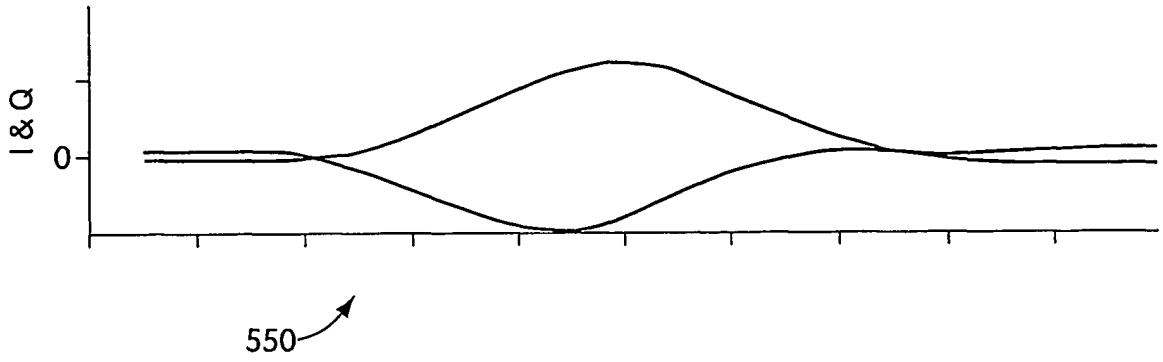



Fig. 5F

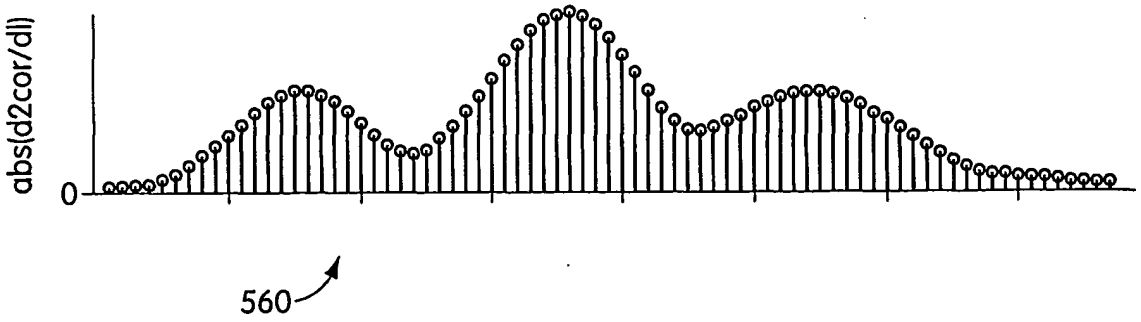



Fig. 5G

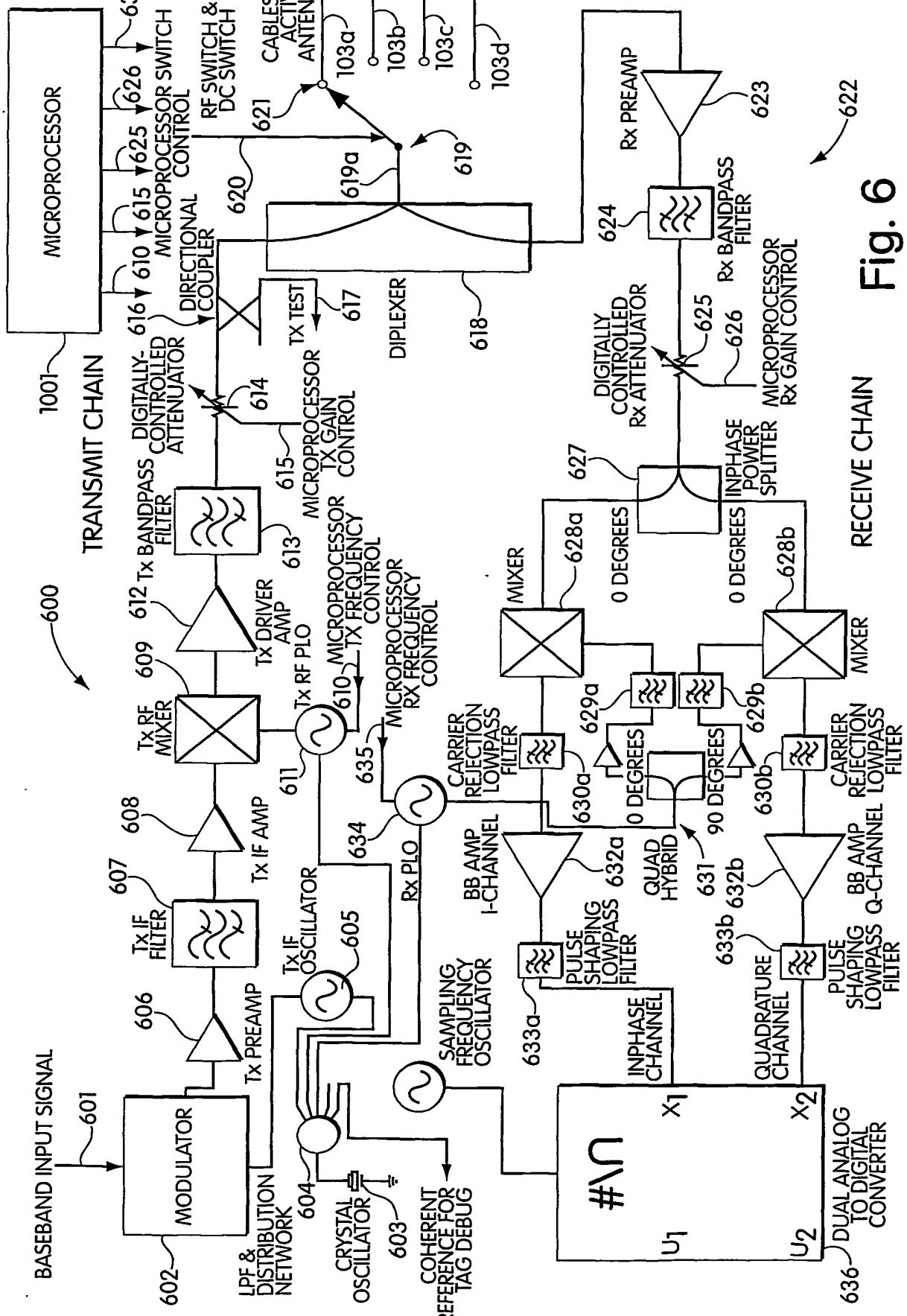

TRANSMIT CHAIN
RECEIVE CHAIN
MICROPROCESSOR
MICROPROCESSOR SWITCH CONTROL
RF SWITCH & DC SWITCH
CABLES TO ACTIVE ANTENNAS
DIRECTIONAL COUPLER
DIPLEXER
TX TEST
DIGITALLY-CONTROLLED ATTENUATOR
MICROPROCESSOR TX GAIN CONTROL
Tx BANDPASS FILTER
Tx DRIVER AMP
MICROPROCESSOR TX FREQUENCY CONTROL
Tx RF PLO
MICROPROCESSOR RX FREQUENCY CONTROL
Tx RF MIXER
Tx IF AMP
Tx IF FILTER
Tx IF OSCILLATOR
Tx PREAMP
SAMPLING FREQUENCY OSCILLATOR
Rx PLO
CARRIER REJECTION LOWPASS FILTER
BB AMP I-CHANNEL
BB AMP Q-CHANNEL
QUAD HYBRID
0 DEGREES
90 DEGREES
PULSE SHAPING LOWPASS FILTER
INPHASE CHANNEL
QUADRATURE CHANNEL
DUAL ANALOG TO DIGITAL CONVERTER
BASEBAND INPUT SIGNAL
MODULATOR
LPF & DISTRIBUTION NETWORK
CRYSTAL OSCILLATOR
COHERENT REFERENCE FOR TAG DEBUG
Rx PREAMP
Rx BANDPASS FILTER
DIGITALLY CONTROLLED Rx ATTENUATOR
MICROPROCESSOR Rx GAIN CONTROL
INPHASE POWER SPLITTER
MIXER
1001
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
619a
620
621
622
623
624
625
626
627
628a
628b
629a
629b
630a
630b
631
632a
632b
633a
633b
634
635
636
U1
U2
X1
X2
103a
103b
103c
103d

Fig. 6

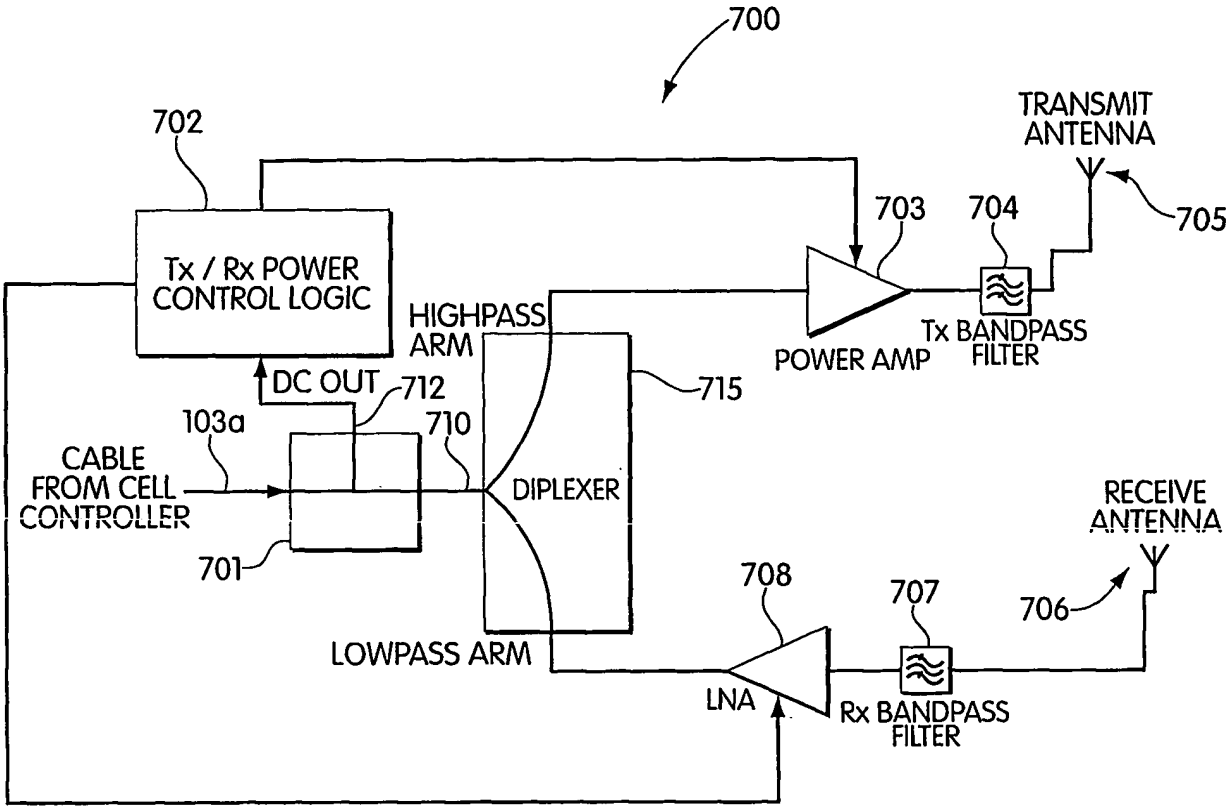

700
702
TRANSMIT ANTENNA
705
703
704
Tx / Rx POWER CONTROL LOGIC
HIGHPASS ARM
POWER AMP
Tx BANDPASS FILTER
DC OUT
712
715
103a
710
CABLE FROM CELL CONTROLLER
DIPLEXER
RECEIVE ANTENNA
701
708
707
706
LOWPASS ARM
LNA
Rx BANDPASS FILTER

Fig. 7

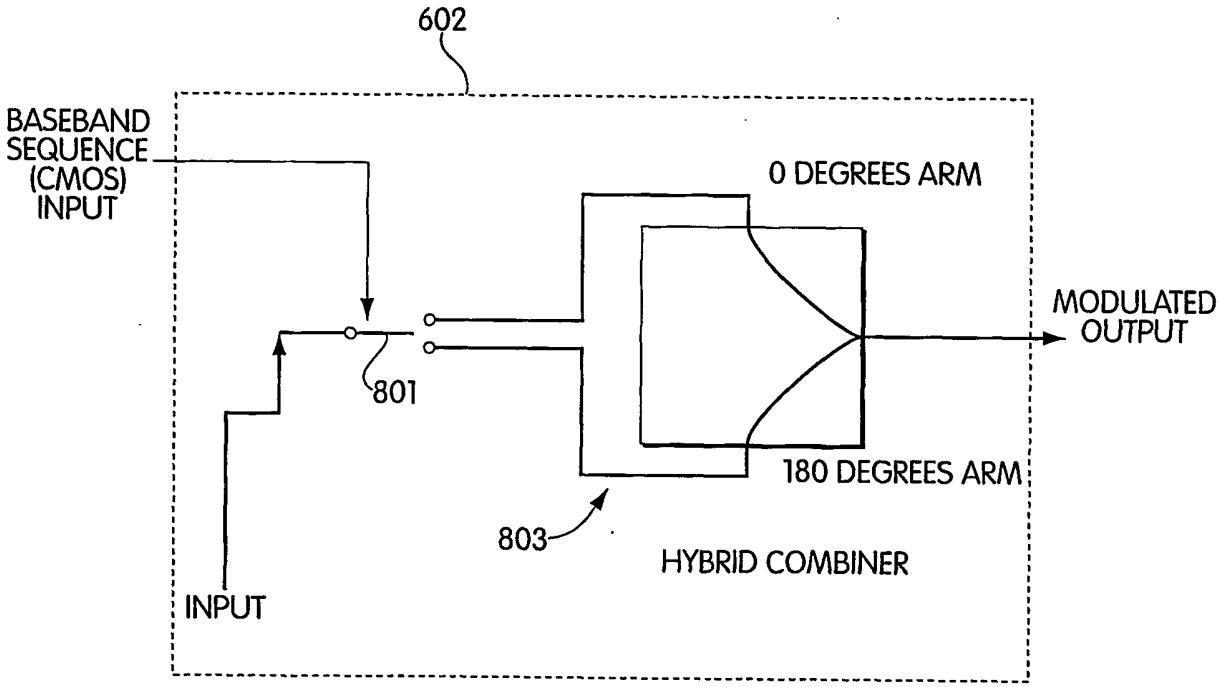

602
BASEBAND SEQUENCE (CMOS) INPUT
0 DEGREES ARM
MODULATED OUTPUT
801
180 DEGREES ARM
803
HYBRID COMBINER
INPUT

Fig. 8

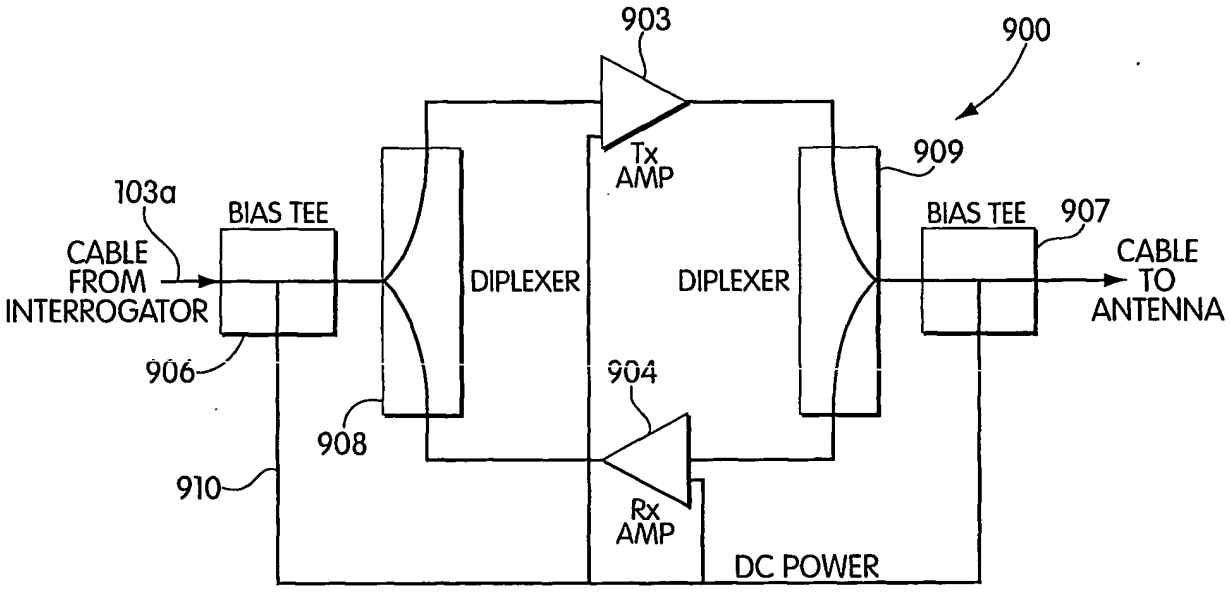

903
900
Tx
AMP
909
103a
BIAS TEE
BIAS TEE
907
CABLE
FROM
INTERROGATOR
DIPLEXER
DIPLEXER
CABLE
TO
ANTENNA
906
904
908
910
Rx
AMP
DC POWER

Fig. 9

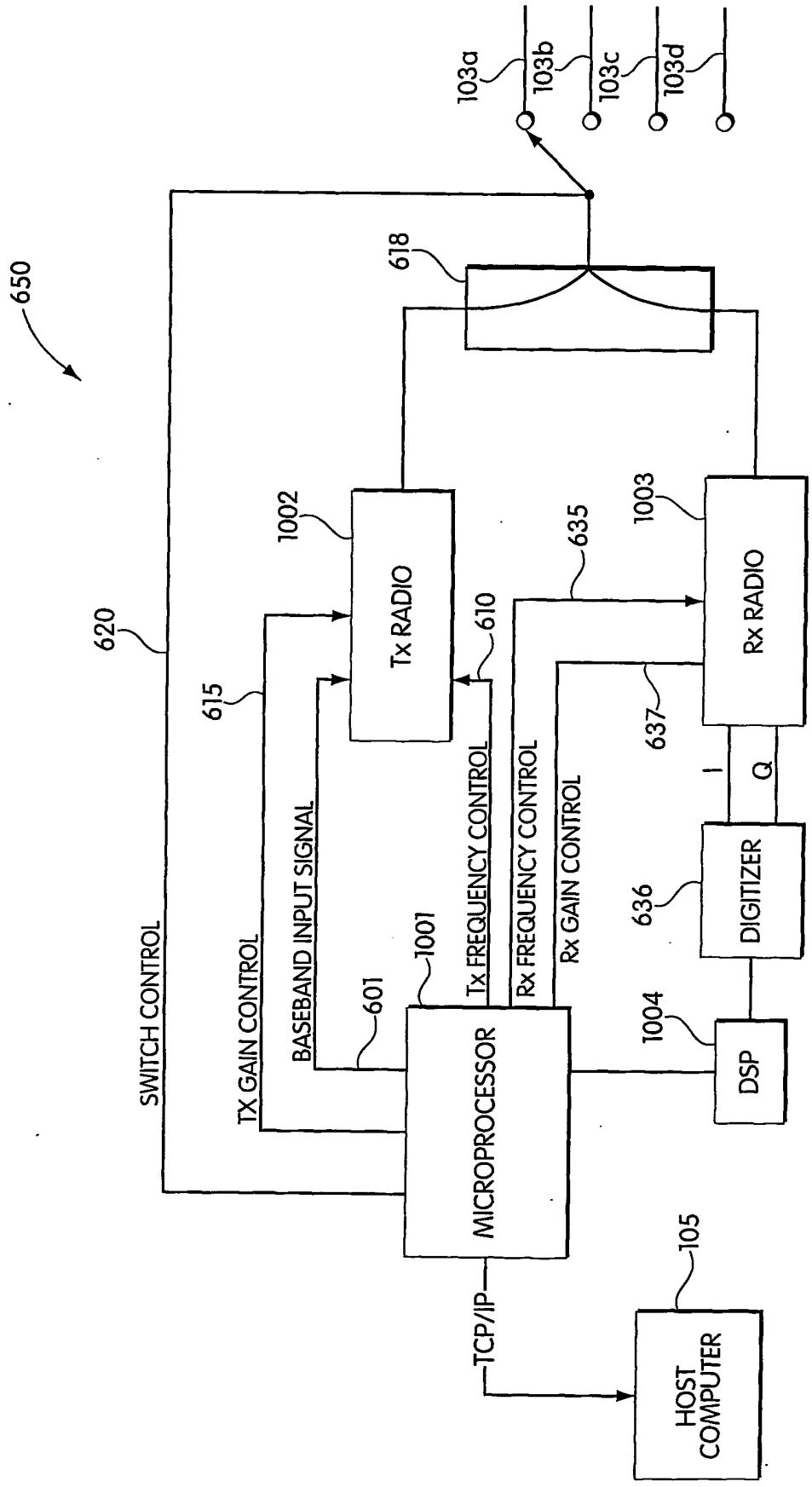

650
103a
103b
103c
103d
618
620
1002
Tx RADIO
610
615
635
1003
Rx RADIO
637
I
Q
636
DIGITIZER
SWITCH CONTROL
TX GAIN CONTROL
BASEBAND INPUT SIGNAL
601
1001
Tx FREQUENCY CONTROL
Rx FREQUENCY CONTROL
Rx GAIN CONTROL
MICROPROCESSOR
1004
DSP
TCP/IP
105
HOST COMPUTER

Fig. 10

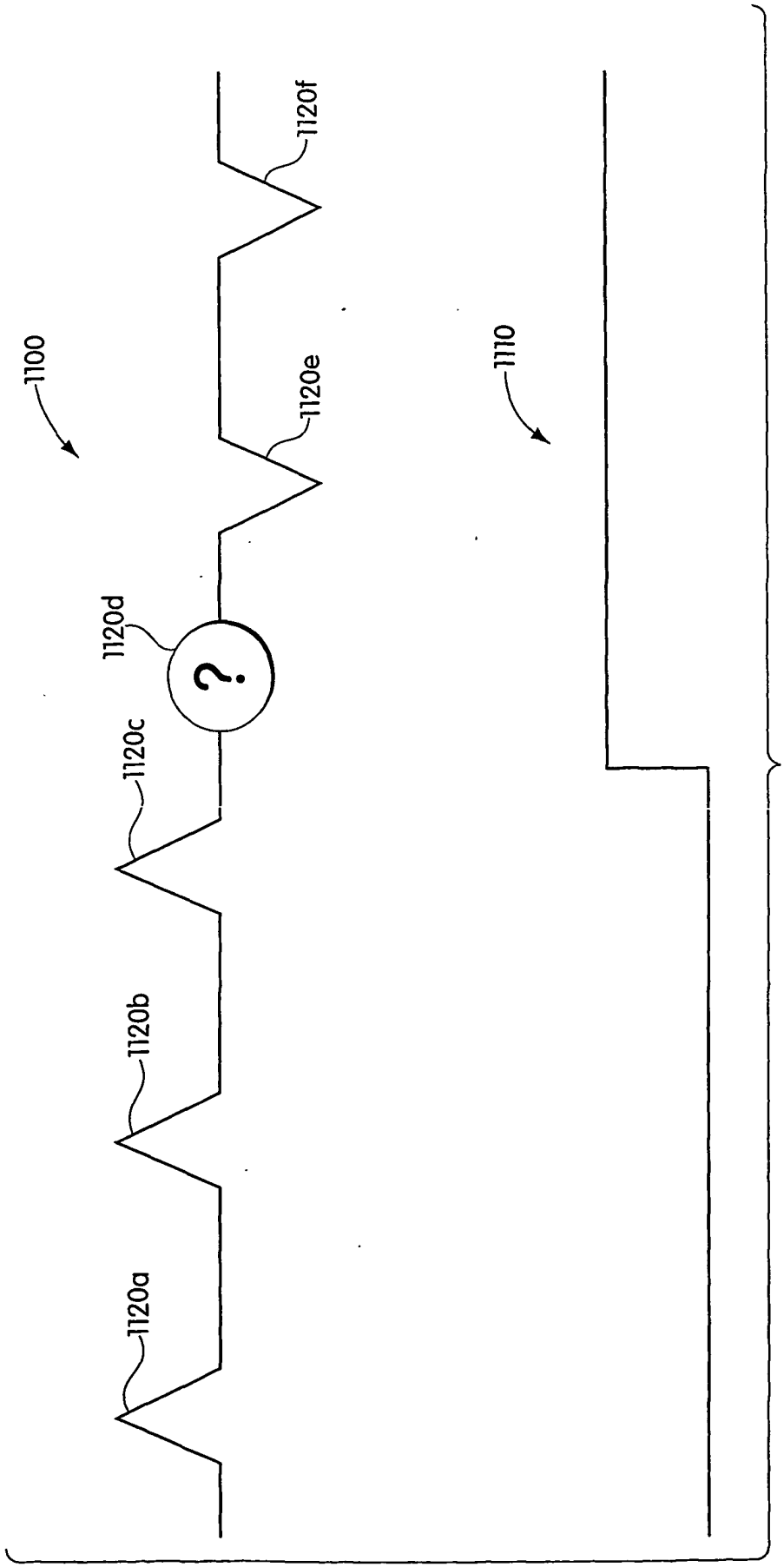



Fig. 11

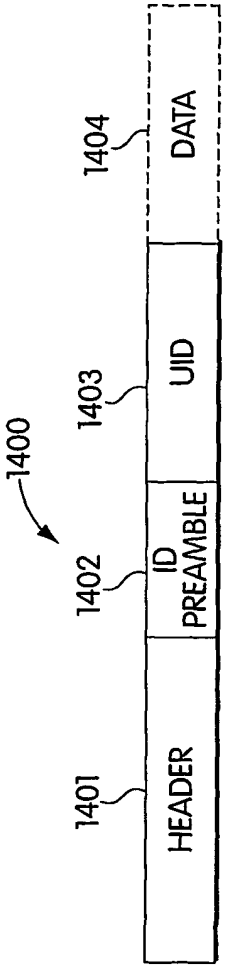



Fig. 12A

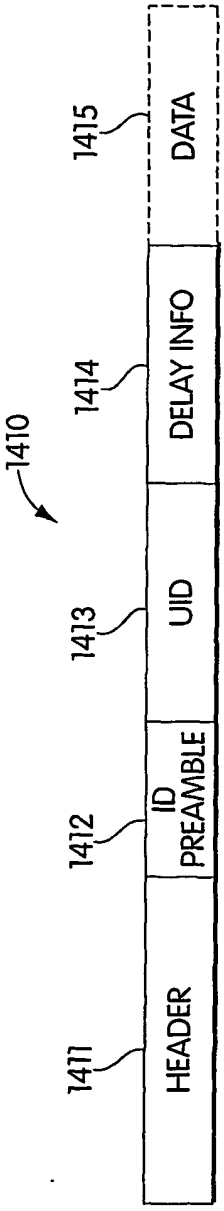



Fig. 12B

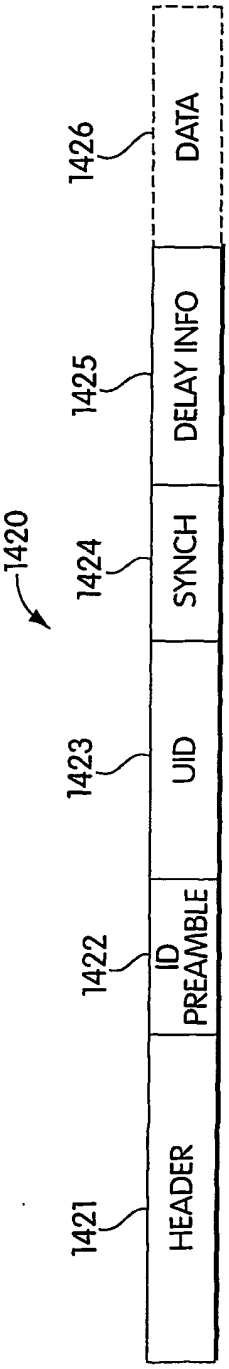



Fig. 12C

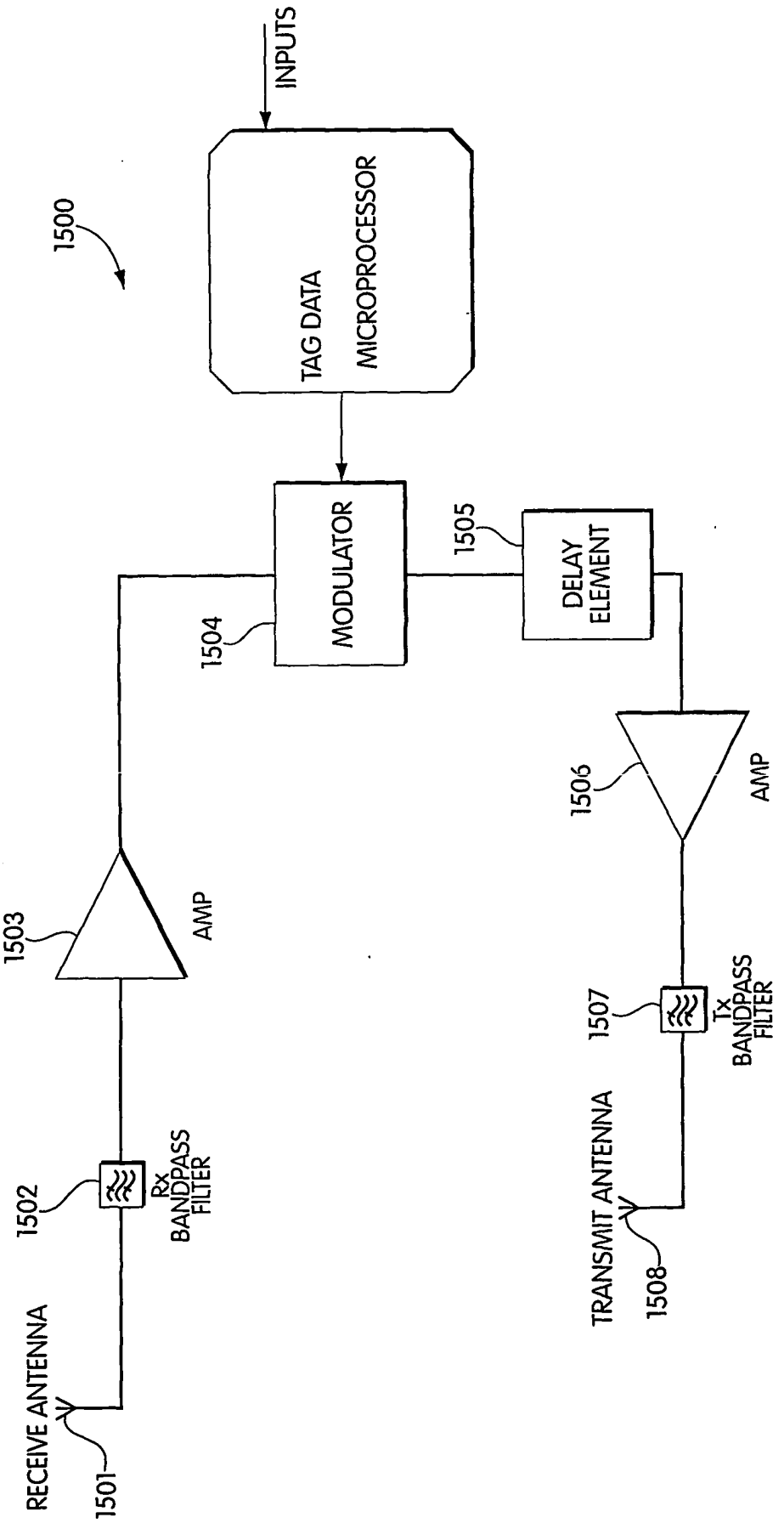

INPUTS
1500
TAG DATA MICROPROCESSOR
MODULATOR
1504
1505
DELAY ELEMENT
1503
AMP
1506
AMP
1502
Rx BANDPASS FILTER
1507
Tx BANDPASS FILTER
RECEIVE ANTENNA
1501
TRANSMIT ANTENNA
1508

Fig. 13

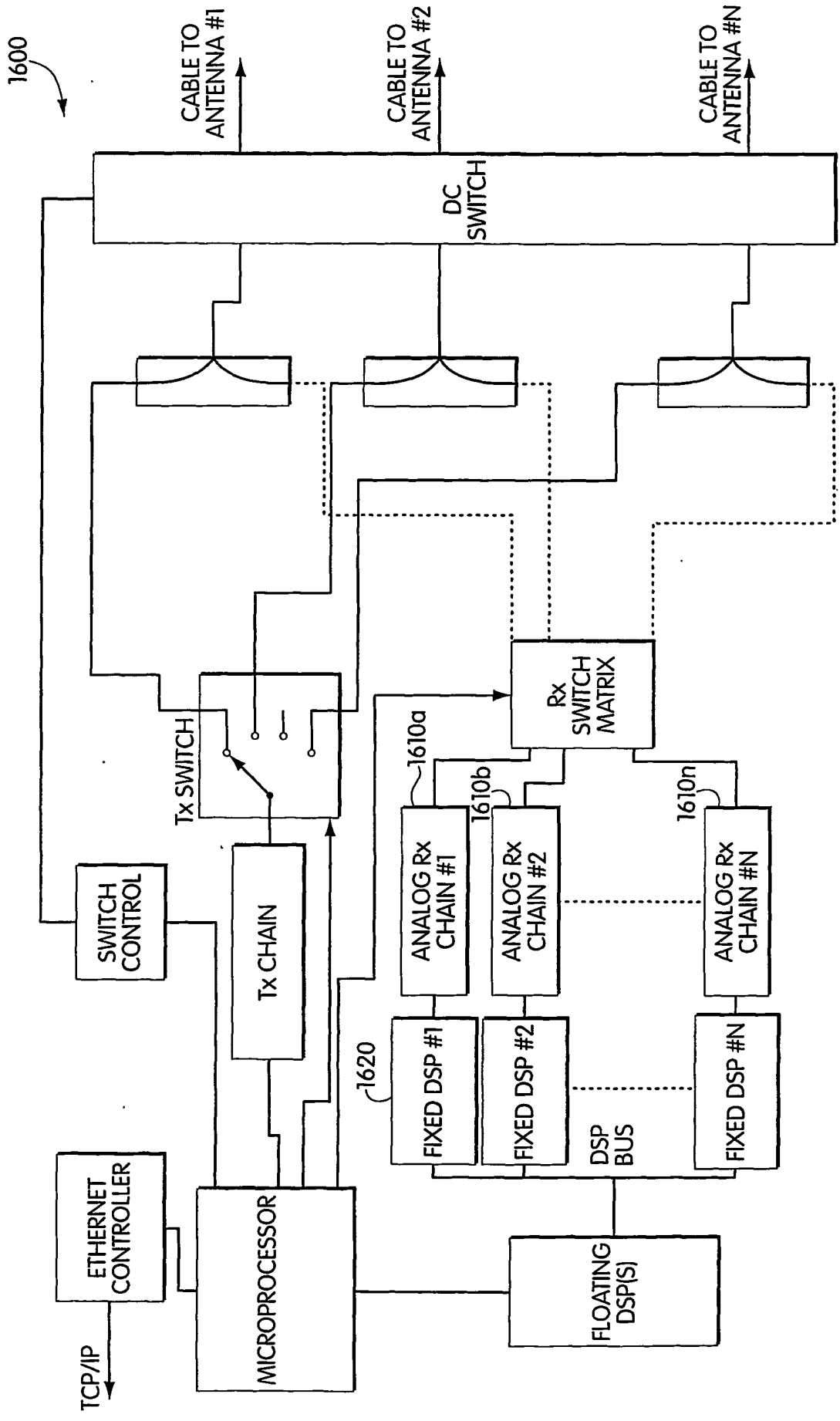

1600
CABLE TO ANTENNA #1
CABLE TO ANTENNA #2
CABLE TO ANTENNA #N
DC SWITCH
Tx SWITCH
Rx SWITCH MATRIX
1610a
1610b
1610n
ANALOG Rx CHAIN #1
ANALOG Rx CHAIN #2
ANALOG Rx CHAIN #N
SWITCH CONTROL
Tx CHAIN
1620
FIXED DSP #1
FIXED DSP #2
FIXED DSP #N
DSP BUS
ETHERNET CONTROLLER
MICROPROCESSOR
FLOATING DSP(S)
TCP/IP

Fig. 14

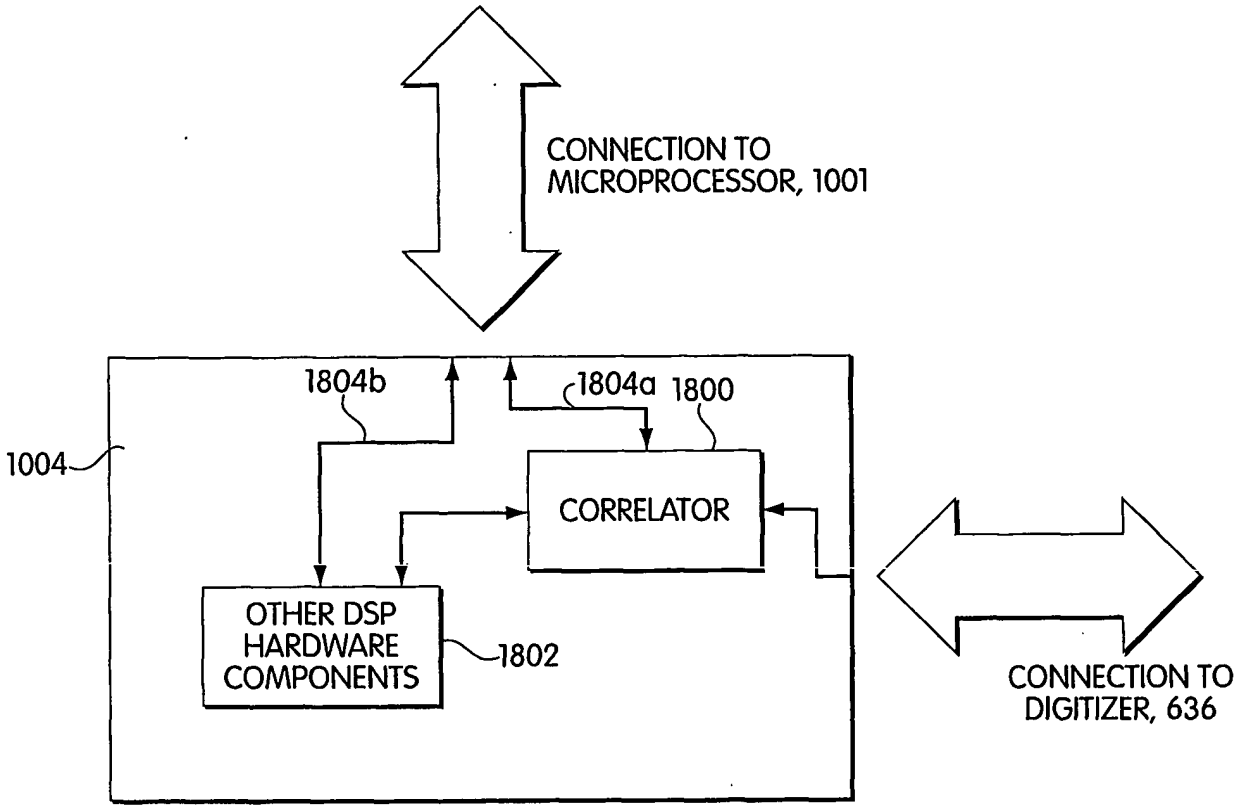

CONNECTION TO
MICROPROCESSOR, 1001
1804b
1804a
1800
1004
CORRELATOR
OTHER DSP
HARDWARE
COMPONENTS
1802
CONNECTION TO
DIGITIZER, 636

Fig. 15A

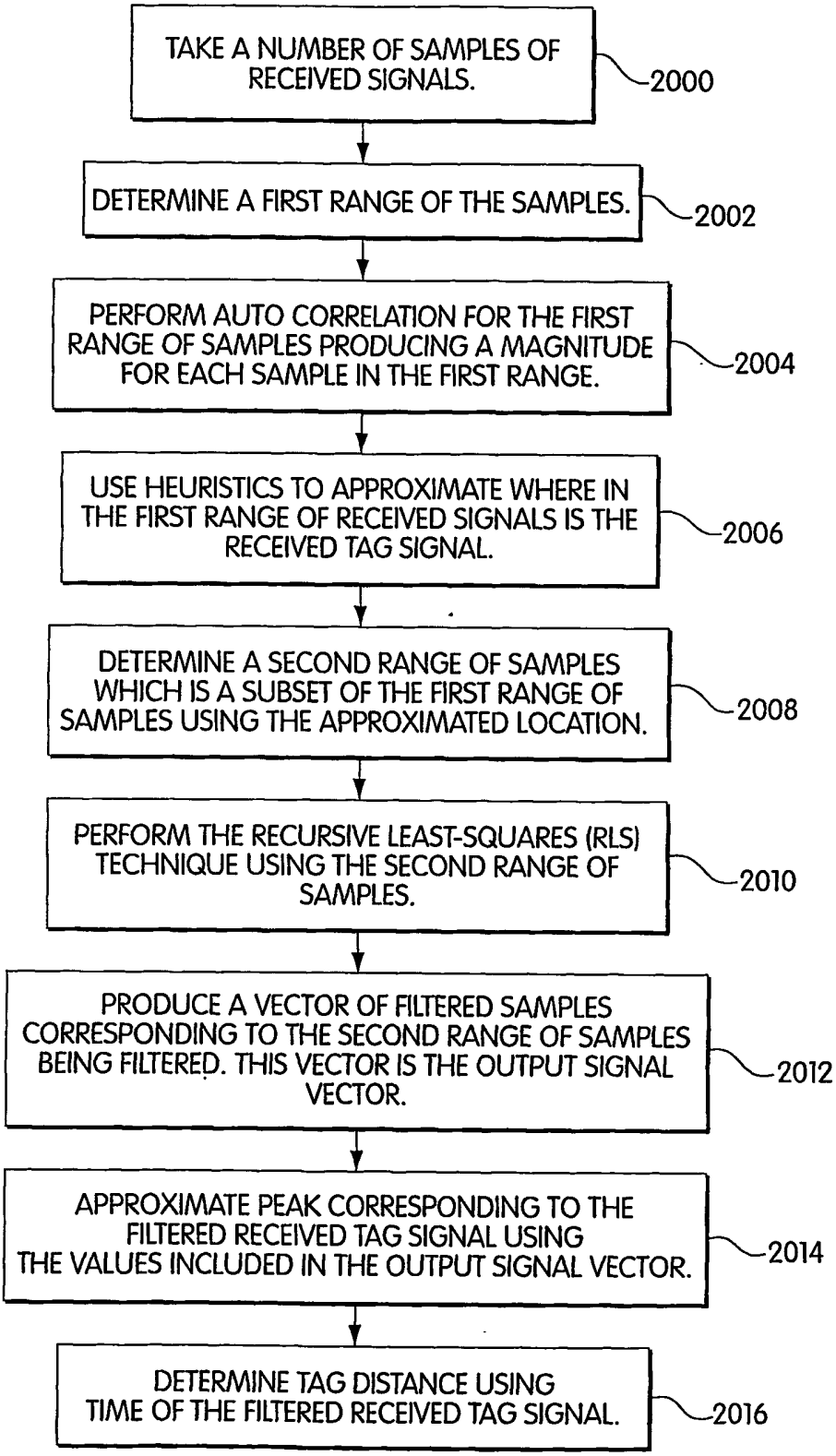

TAKE A NUMBER OF SAMPLES OF RECEIVED SIGNALS.
2000
DETERMINE A FIRST RANGE OF THE SAMPLES.
2002
PERFORM AUTO CORRELATION FOR THE FIRST RANGE OF SAMPLES PRODUCING A MAGNITUDE FOR EACH SAMPLE IN THE FIRST RANGE.
2004
USE HEURISTICS TO APPROXIMATE WHERE IN THE FIRST RANGE OF RECEIVED SIGNALS IS THE RECEIVED TAG SIGNAL.
2006
DETERMINE A SECOND RANGE OF SAMPLES WHICH IS A SUBSET OF THE FIRST RANGE OF SAMPLES USING THE APPROXIMATED LOCATION.
2008
PERFORM THE RECURSIVE LEAST-SQUARES (RLS) TECHNIQUE USING THE SECOND RANGE OF SAMPLES.
2010
PRODUCE A VECTOR OF FILTERED SAMPLES CORRESPONDING TO THE SECOND RANGE OF SAMPLES BEING FILTERED. THIS VECTOR IS THE OUTPUT SIGNAL VECTOR.
2012
APPROXIMATE PEAK CORRESPONDING TO THE FILTERED RECEIVED TAG SIGNAL USING THE VALUES INCLUDED IN THE OUTPUT SIGNAL VECTOR.
2014
DETERMINE TAG DISTANCE USING TIME OF THE FILTERED RECEIVED TAG SIGNAL.
2016

Fig. 15B

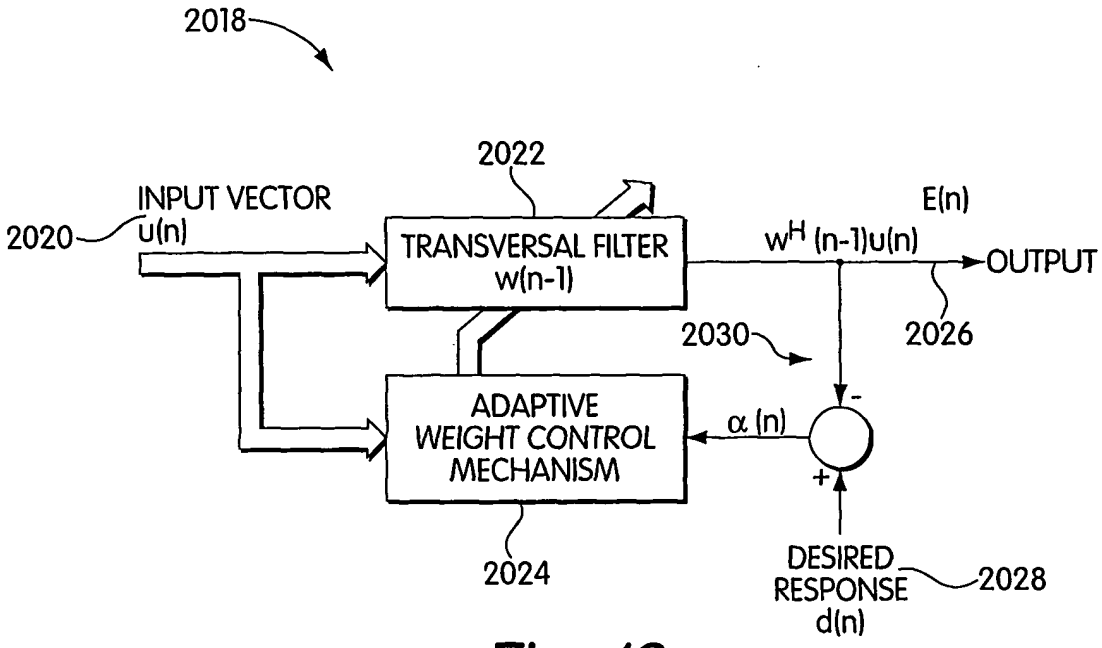

2018
2022
INPUT VECTOR
u(n)
2020
TRANSVERSAL FILTER
w(n-1)
E(n)
$w^H$ (n-1)u(n)
OUTPUT
2026
2030
ADAPTIVE
WEIGHT CONTROL
MECHANISM
α (n)
-
+
2024
DESIRED
RESPONSE
d(n)
2028

Fig. 16

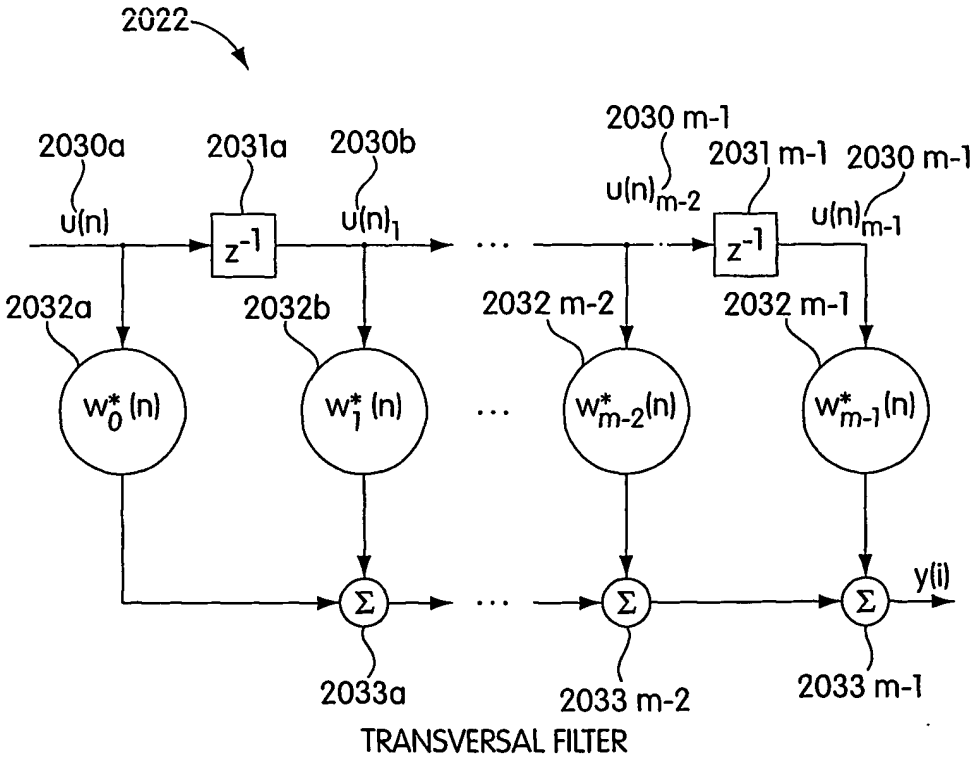

2022
2030a
2031a
2030b
2030 m-1
2031 m-1
2030 m-1
u(n)
$u(n)_1$
$u(n)_{m-2}$
$u(n)_{m-1}$
$z^{-1}$
2032a
2032b
2032 m-2
2032 m-1
$w_0^*$ (n)
$w_1^*$ (n)
$w_{m-2}^*$(n)
$w_{m-1}^*$(n)
Σ
y(i)
2033a
2033 m-2
2033 m-1
TRANSVERSAL FILTER

Fig. 17

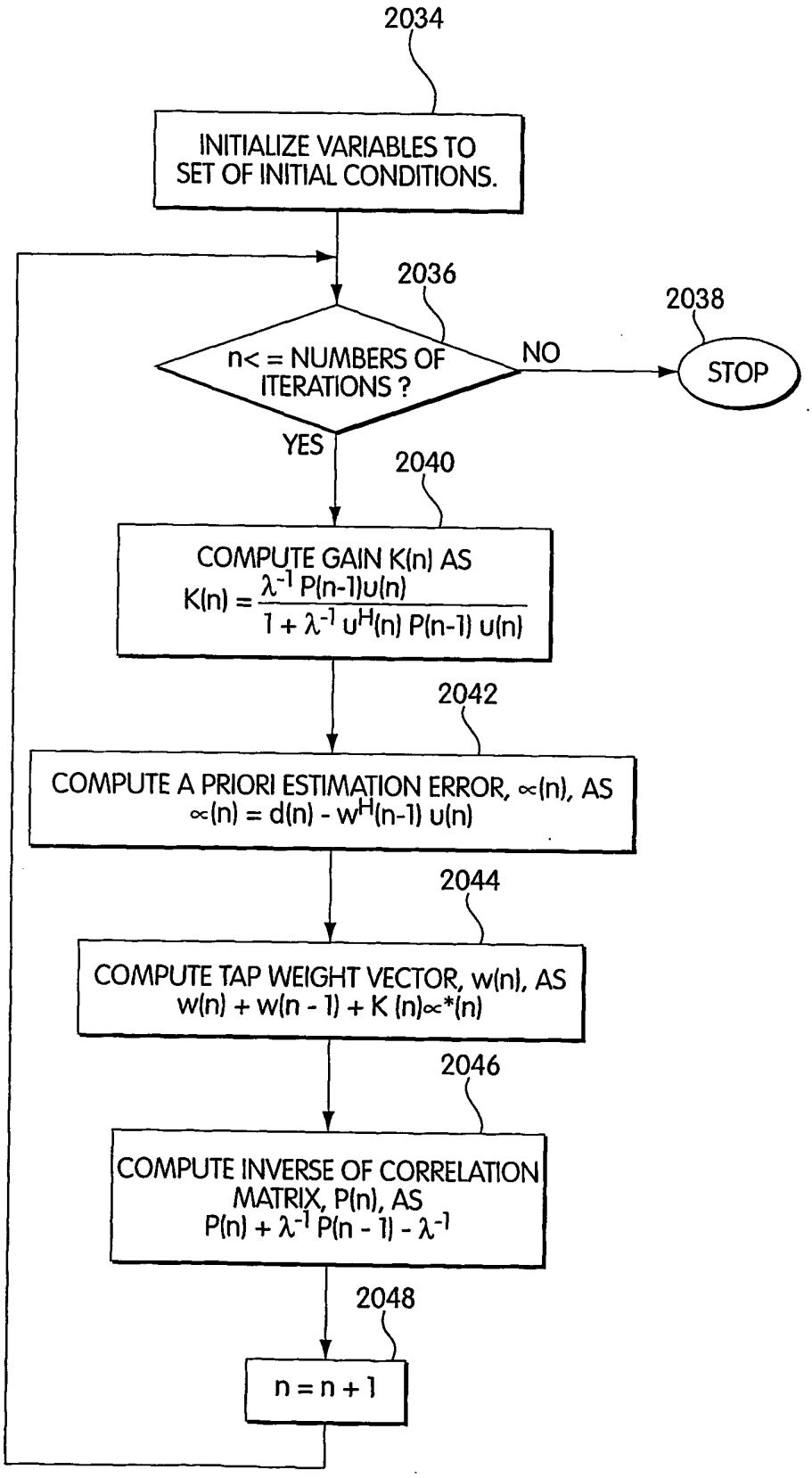

2034
INITIALIZE VARIABLES TO SET OF INITIAL CONDITIONS.
2036
n< = NUMBERS OF ITERATIONS ?
NO
2038
STOP
YES
2040
COMPUTE GAIN K(n) AS
$K(n) = \frac{\lambda^{-1} P(n-1)u(n)}{1 + \lambda^{-1} u^H(n) P(n-1) u(n)}$
2042
COMPUTE A PRIORI ESTIMATION ERROR, ∝(n), AS
∝(n) = d(n) - $w^H$(n-1) u(n)
2044
COMPUTE TAP WEIGHT VECTOR, w(n), AS
w(n) + w(n - 1) + K (n)∝*(n)
2046
COMPUTE INVERSE OF CORRELATION MATRIX, P(n), AS
P(n) + $\lambda^{-1}$ P(n - 1) - $\lambda^{-1}$
2048
n = n + 1

Fig. 18

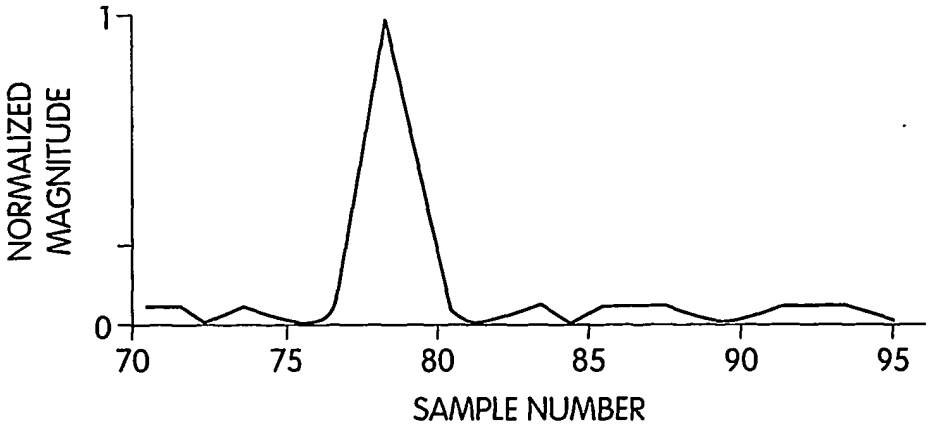



Fig. 19A

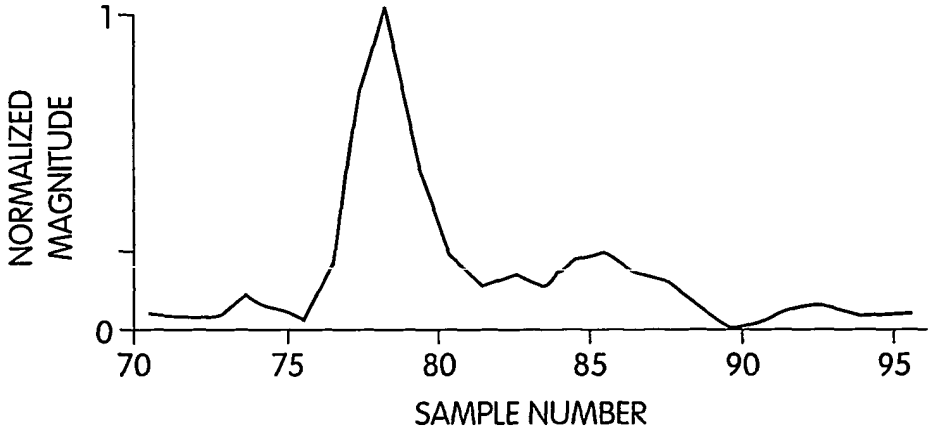



Fig. 19B

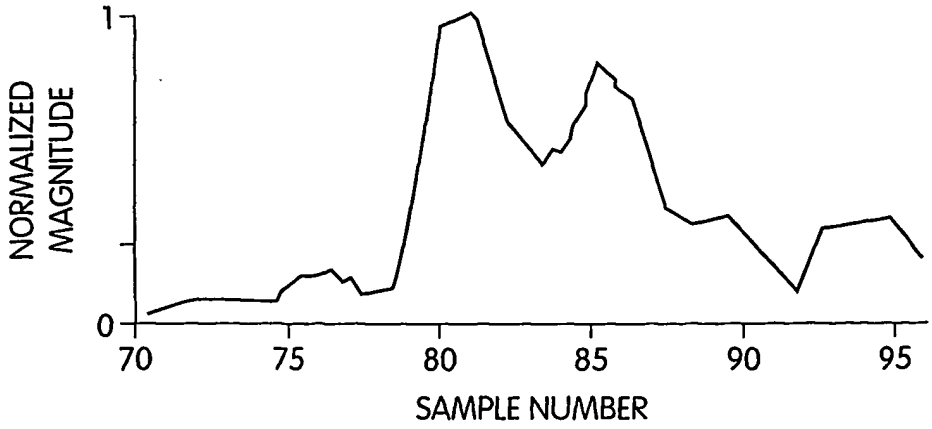



Fig. 19C

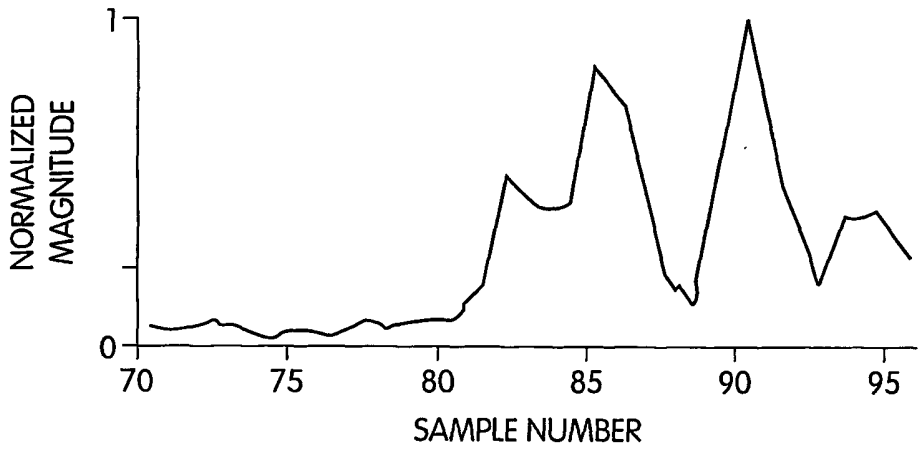



Fig. 19D

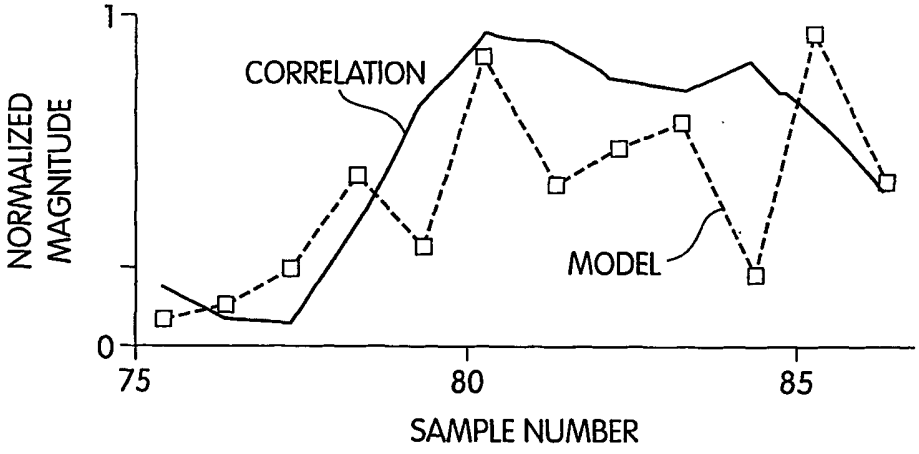



Fig. 19E

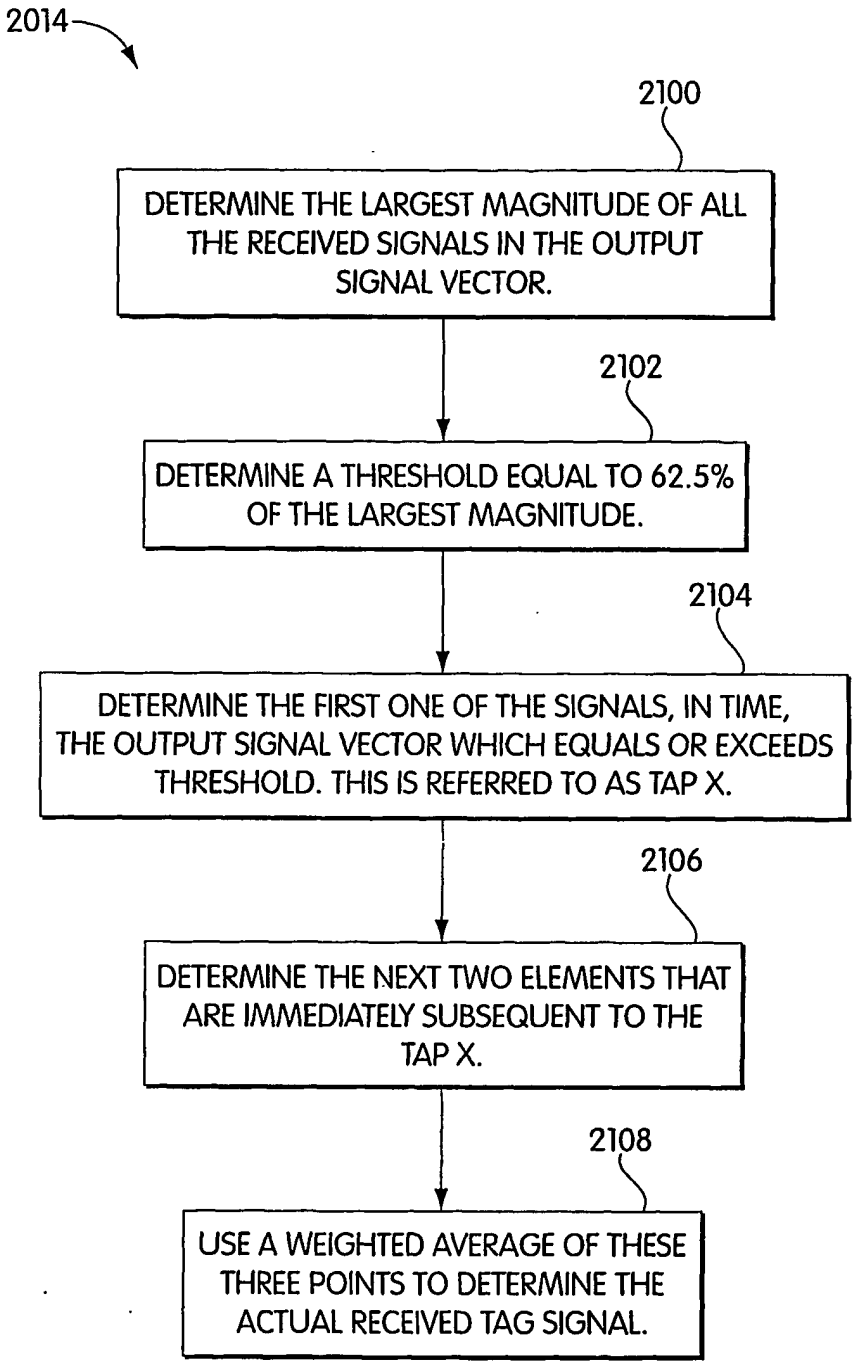

2014
2100
DETERMINE THE LARGEST MAGNITUDE OF ALL THE RECEIVED SIGNALS IN THE OUTPUT SIGNAL VECTOR.
2102
DETERMINE A THRESHOLD EQUAL TO 62.5% OF THE LARGEST MAGNITUDE.
2104
DETERMINE THE FIRST ONE OF THE SIGNALS, IN TIME, THE OUTPUT SIGNAL VECTOR WHICH EQUALS OR EXCEEDS THRESHOLD. THIS IS REFERRED TO AS TAP X.
2106
DETERMINE THE NEXT TWO ELEMENTS THAT ARE IMMEDIATELY SUBSEQUENT TO THE TAP X.
2108
USE A WEIGHTED AVERAGE OF THESE THREE POINTS TO DETERMINE THE ACTUAL RECEIVED TAG SIGNAL.

Fig. 20

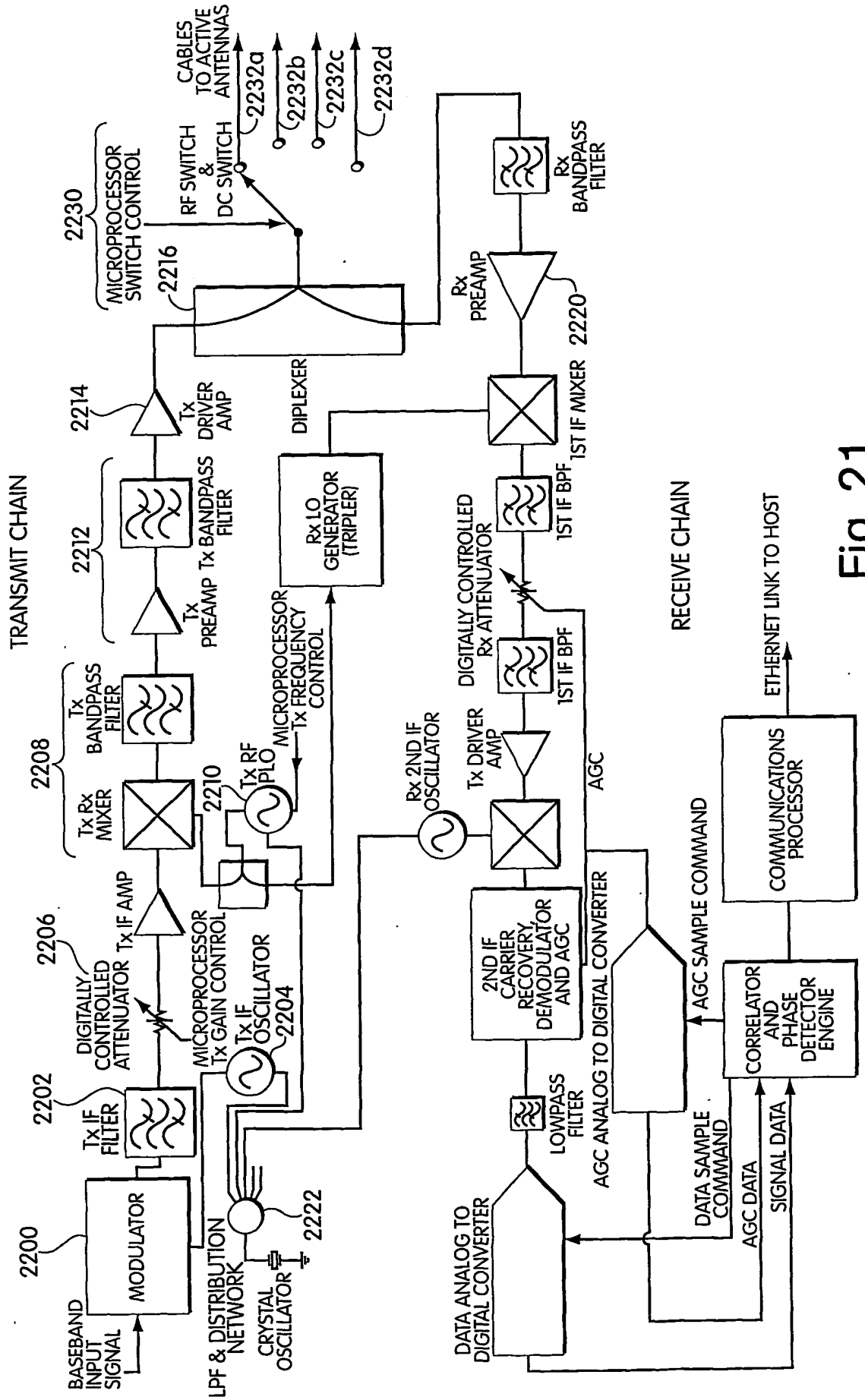

TRANSMIT CHAIN
RECEIVE CHAIN
BASEBAND INPUT SIGNAL
2200
MODULATOR
2202
Tx IF FILTER
DIGITALLY CONTROLLED ATTENUATOR
2206
Tx IF AMP
MICROPROCESSOR Tx GAIN CONTROL
Tx IF OSCILLATOR 2204
LPF & DISTRIBUTION NETWORK
CRYSTAL OSCILLATOR
2222
2208
Tx Rx MIXER
Tx BANDPASS FILTER
2210
Tx RF PLO
MICROPROCESSOR Tx FREQUENCY CONTROL
2212
Tx PREAMP
Tx BANDPASS FILTER
2214
Tx DRIVER AMP
2216
MICROPROCESSOR SWITCH CONTROL
2230
RF SWITCH & DC SWITCH
CABLES TO ACTIVE ANTENNAS
2232a
2232b
2232c
2232d
DIPLEXER
Rx LO GENERATOR (TRIPLER)
Rx 2ND IF OSCILLATOR
Tx DRIVER AMP
DIGITALLY CONTROLLED Rx ATTENUATOR
1ST IF BPF
1ST IF BPF
1ST IF MIXER
2220
Rx PREAMP
Rx BANDPASS FILTER
AGC
2ND IF CARRIER RECOVERY DEMODULATOR AND AGC
LOWPASS FILTER
AGC ANALOG TO DIGITAL CONVERTER
DATA ANALOG TO DIGITAL CONVERTER
AGC SAMPLE COMMAND
DATA SAMPLE COMMAND
AGC DATA
SIGNAL DATA
CORRELATOR AND PHASE DETECTOR ENGINE
COMMUNICATIONS PROCESSOR
ETHERNET LINK TO HOST

Fig. 21

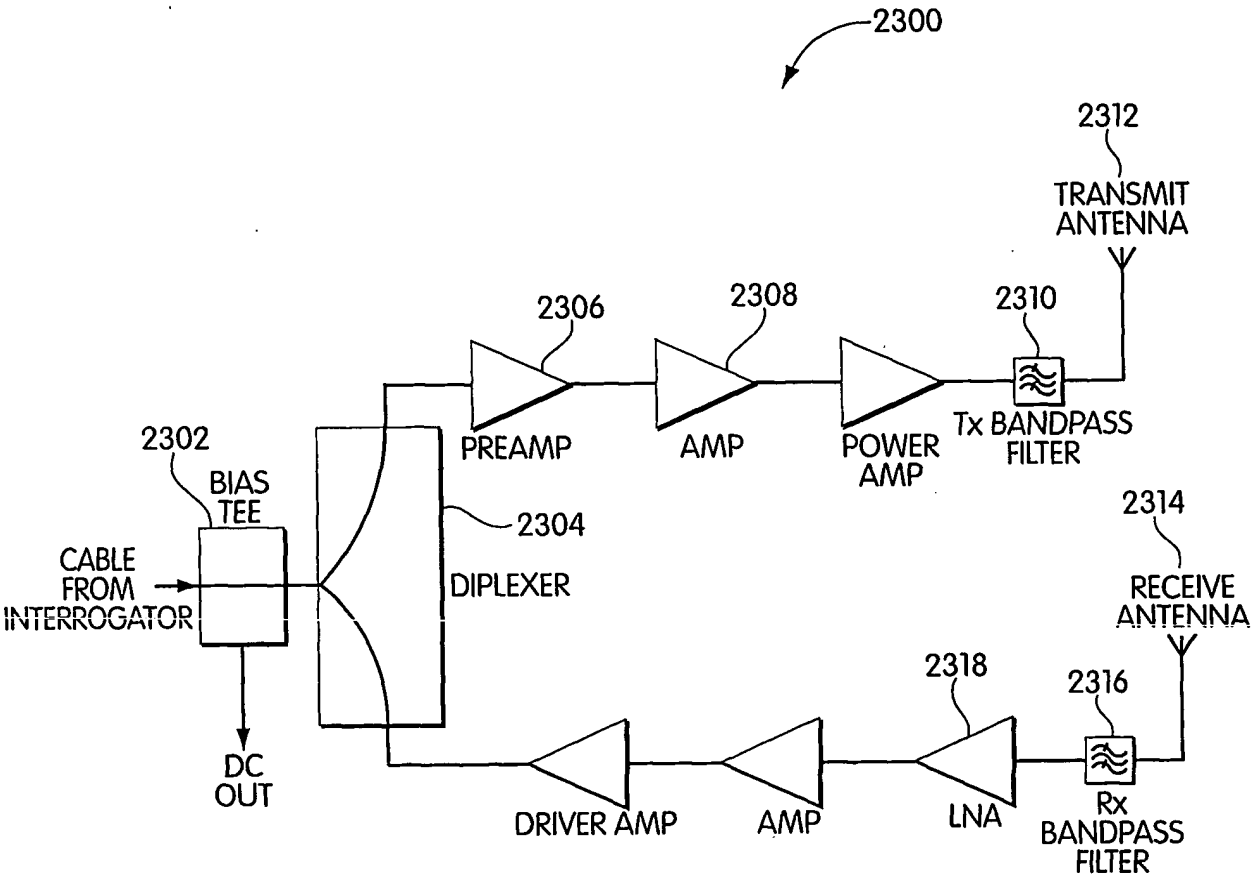

2300
2312
TRANSMIT ANTENNA
2306
2308
2310
PREAMP
AMP
POWER AMP
Tx BANDPASS FILTER
2302
BIAS TEE
2304
2314
CABLE FROM INTERROGATOR
DIPLEXER
RECEIVE ANTENNA
2318
2316
DC OUT
DRIVER AMP
AMP
LNA
Rx BANDPASS FILTER

Fig. 22

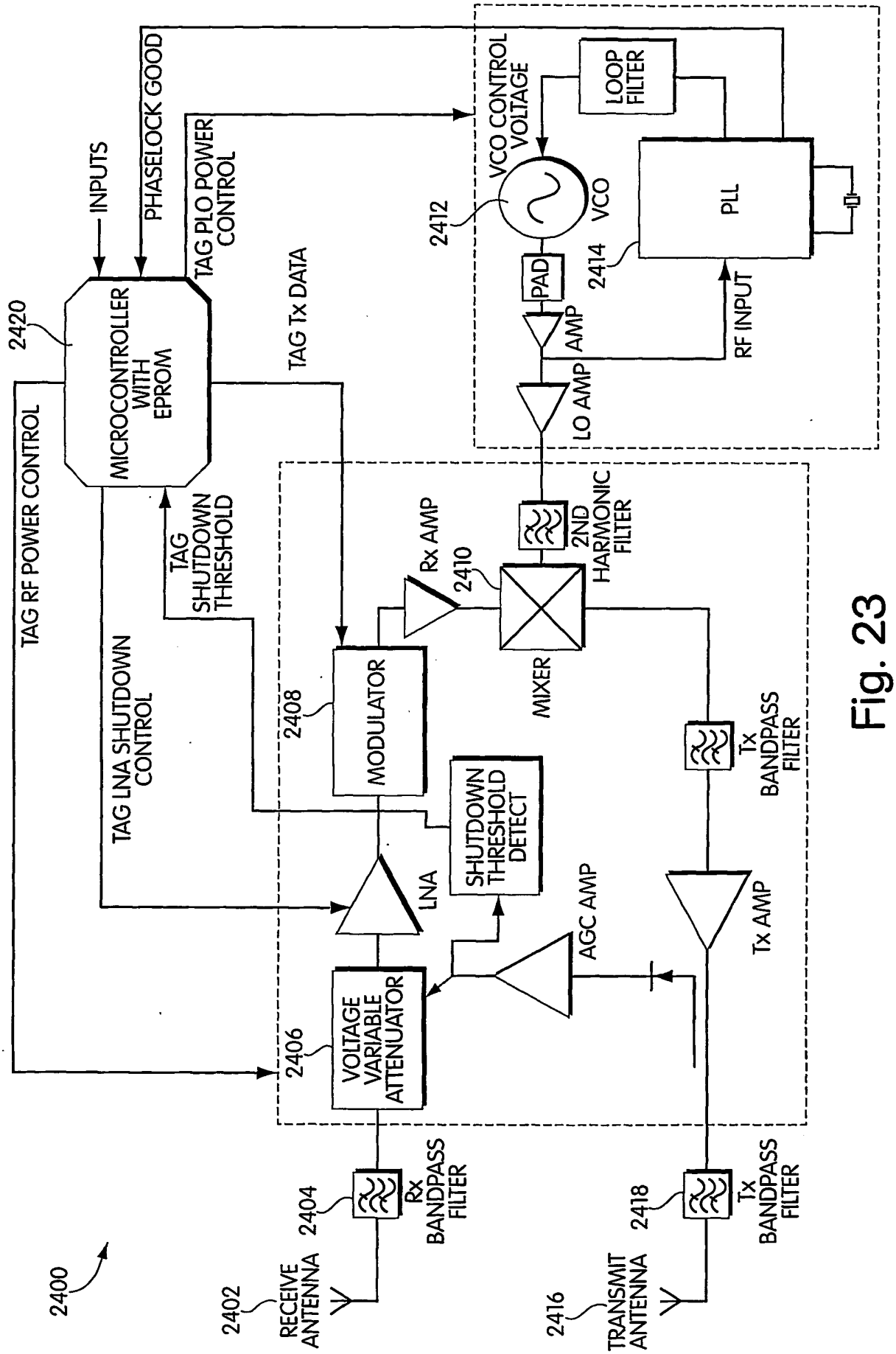

2400
2402
RECEIVE ANTENNA
2404
Rx BANDPASS FILTER
2406
VOLTAGE VARIABLE ATTENUATOR
2408
MODULATOR
2410
MIXER
2412
VCO
2414
PLL
2416
TRANSMIT ANTENNA
2418
Tx BANDPASS FILTER
2420
MICROCONTROLLER WITH EPROM
INPUTS
PHASELOCK GOOD
TAG PLO POWER CONTROL
TAG Tx DATA
TAG RF POWER CONTROL
TAG LNA SHUTDOWN CONTROL
TAG SHUTDOWN THRESHOLD
VCO CONTROL VOLTAGE
LOOP FILTER
PAD
AMP
LO AMP
RF INPUT
Rx AMP
2ND HARMONIC FILTER
LNA
SHUTDOWN THRESHOLD DETECT
AGC AMP
Tx AMP
Tx BANDPASS FILTER

Fig. 23

| PARAMETER | MIN. VALUE | NOM. VALUE | MAX. VALUE | UNITS | COMMENTS |
|---|---|---|---|---|---|
| Rx SIGNAL CENTER FREQUENCY | | 2442 | | MHz | |
| Rx INPUT SIGNAL LEVEL | -45 | | -5 | dBm | |
| Rx SIGNAL BANDWIDTH | | 80 | | MHz | @ -3dB POINTS |
| Rx NOISE FIGURE | | | 6 | dB | |
| Tx SIGNAL CENTER FREQUENCY | | 5800 | | MHz | |
| Tx SIGNAL BANDWIDTH | | 80 | | MHz | @ -3dB POINTS |
| LINEAR RF GAIN | 30 | 35 | | | @ INPUT OF -45dBm WITH NO AGC |
| Tx OUTPUT SIGNAL LEVEL | -10 | | -5 | dBm | WITH NORMAL AGC OPERATION |
| CRYSTAL FREQUENCY | | 6.5000 | | MHz | NOMINAL |
| LO TURN-ON TIME | | | 3.5 | msec | 2msec xtal, 0.75msec PLL |
| MODULATOR PHASE ACCURACY | | | ±3 | ° | |
| MODULATOR AMPLITUDE ACCURACY | | | ±0.5 | dB | |
| CURRENT CONSUMPTION DURING TRANSMIT | | | 55 | ma | INCLUDING 2mA FOR MICROCONTROLLER |
| CURRENT CONSUMPTION DURING SLEEP | | | 40 | µA | RF SHUT DOWN; MICROCONTROLLER SLEEP |
| TAG LINEARITY | | | -50 | dBm | EIRP, RADIATED AT TAG OUTPUT, ±75MHz FROM NOMINAL CENTER FREQUENCY, CONSISTENT WITH FCC PART 15.249, FOR ANY INPUT LEVEL |

Fig. 24

SUBSTITUTE SHEET (RULE 26)

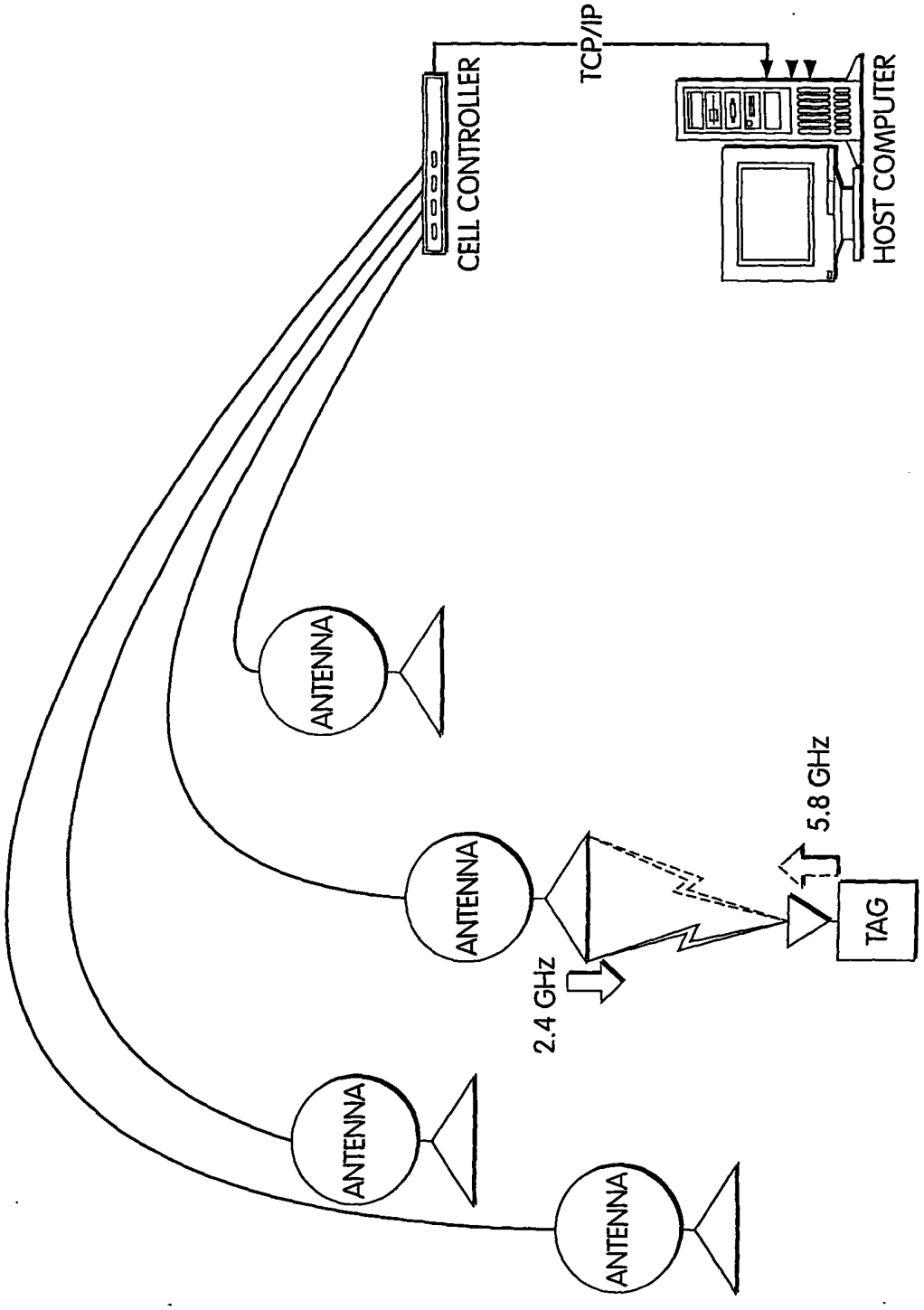

TCP/IP
CELL CONTROLLER
HOST COMPUTER
ANTENNA
ANTENNA
ANTENNA
5.8 GHz
2.4 GHz
TAG

Fig. 25

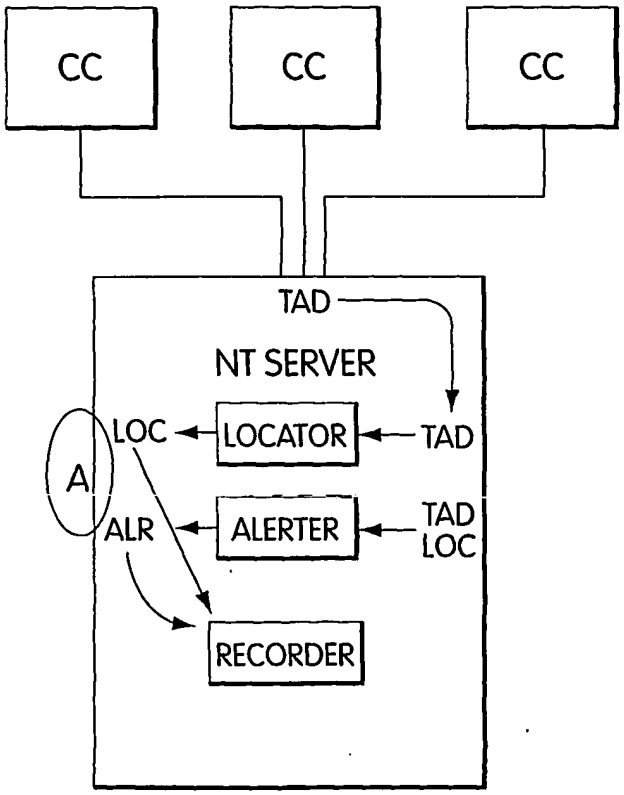

CC
CC
CC
TAD
NT SERVER
LOC
LOCATOR
TAD
A
ALR
ALERTER
TAD
LOC
RECORDER

Fig. 26A

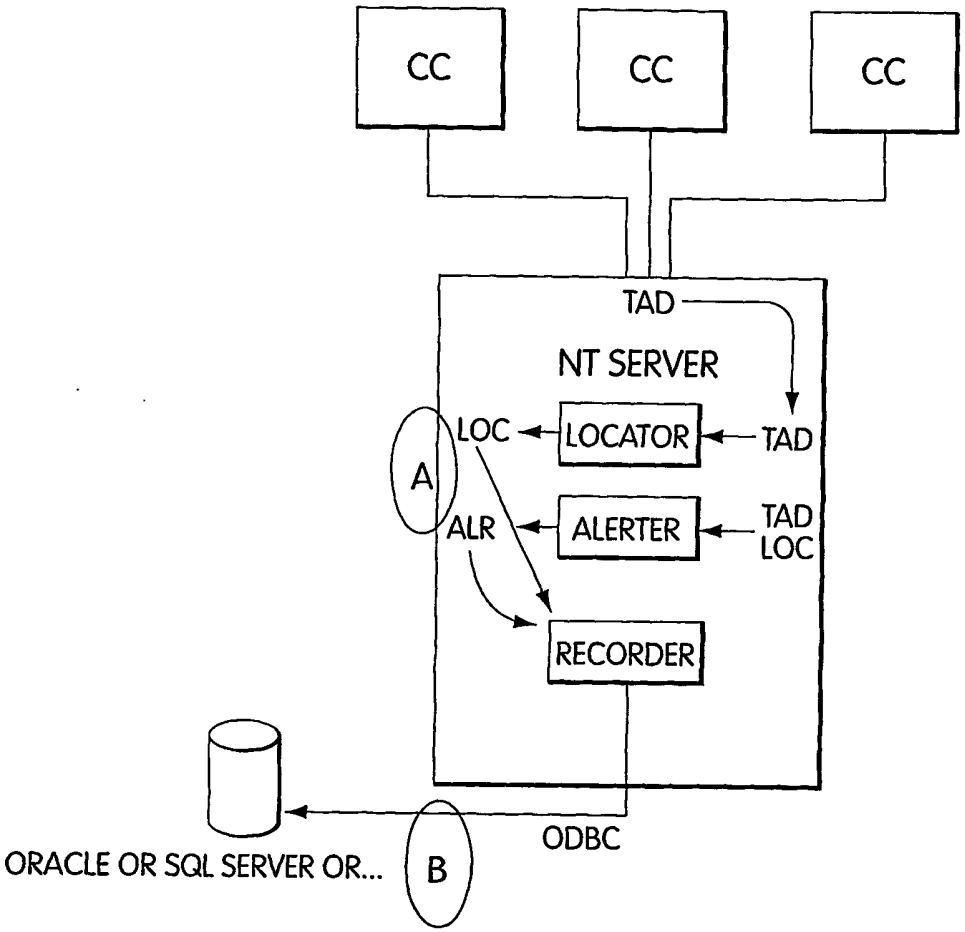

CC
CC
CC
TAD
NT SERVER
LOC
LOCATOR
TAD
A
ALR
ALERTER
TAD
LOC
RECORDER
ODBC
ORACLE OR SQL SERVER OR...
B

Fig. 26B

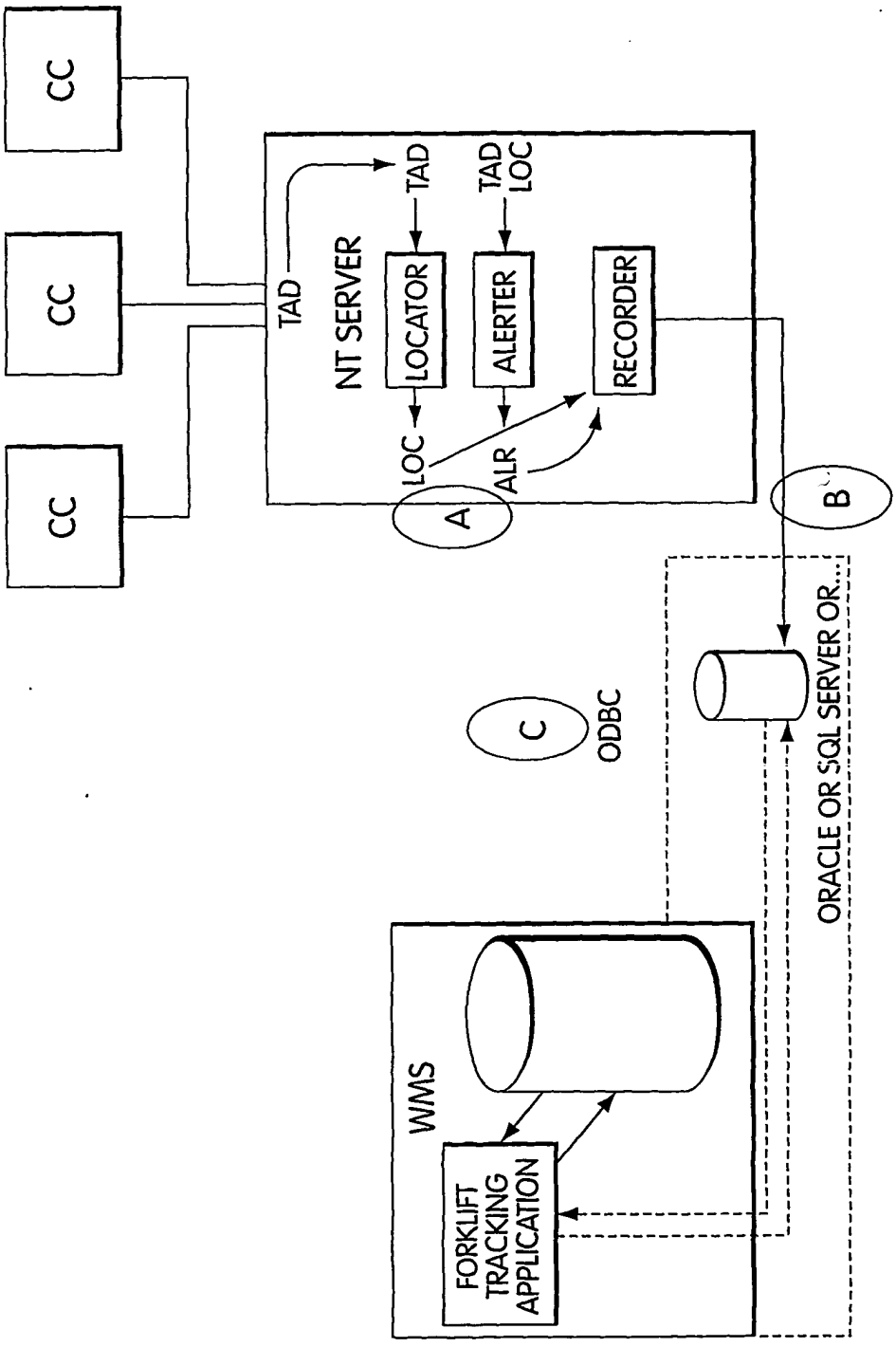

CC
CC
CC
TAD
NT SERVER
TAD
TAD LOC
LOCATOR
ALERTER
RECORDER
LOC
ALR
A
B
C
ODBC
ORACLE OR SQL SERVER OR...
WMS
FORKLIFT TRACKING APPLICATION

Fig. 26C

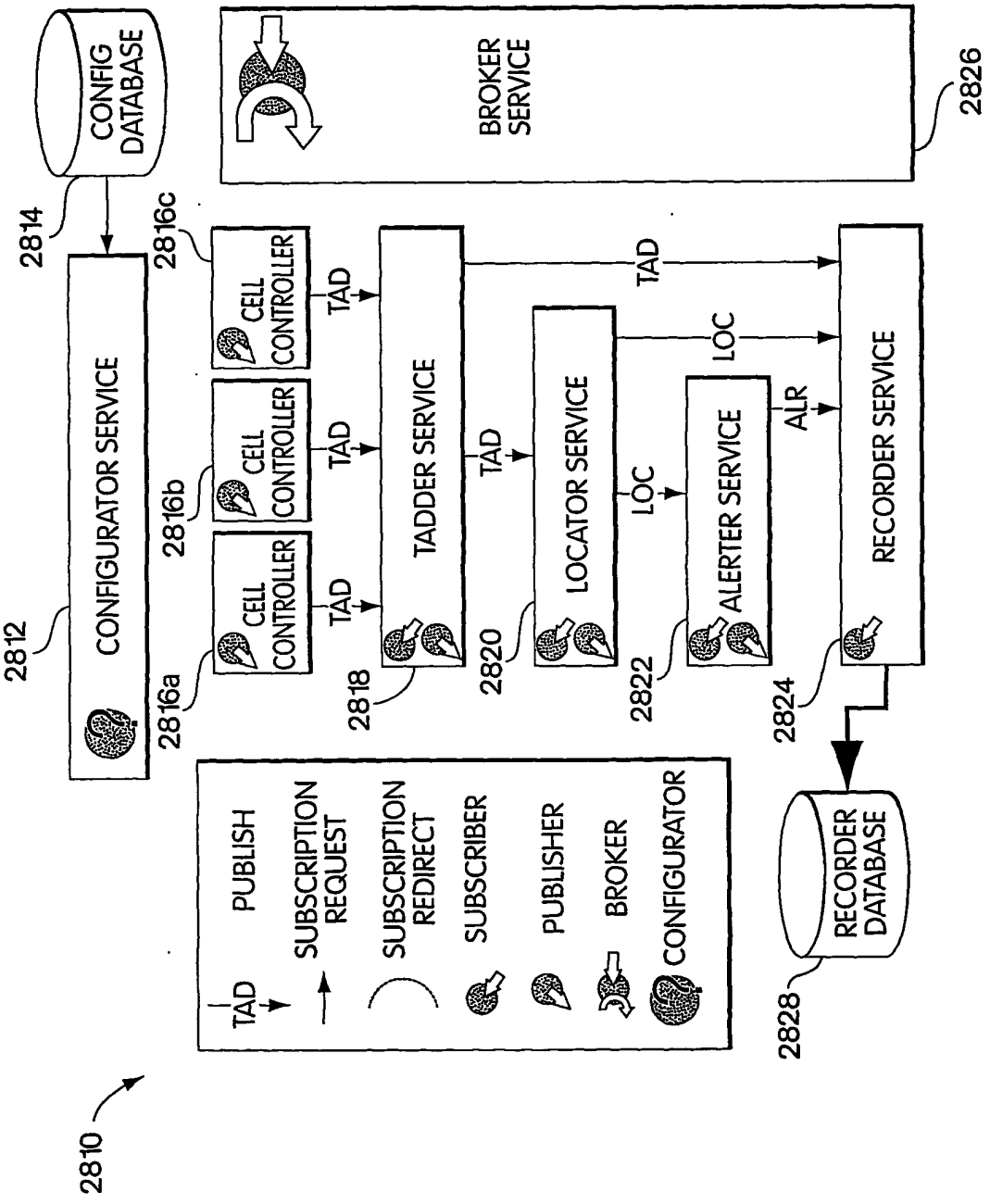

CONFIG DATABASE
2814
BROKER SERVICE
2826
CONFIGURATOR SERVICE
2812
2816a
2816b
2816c
CELL CONTROLLER
CELL CONTROLLER
CELL CONTROLLER
TAD
TAD
TAD
TADDER SERVICE
2818
TAD
TAD
LOCATOR SERVICE
2820
LOC
LOC
ALERTER SERVICE
2822
ALR
RECORDER SERVICE
2824
RECORDER DATABASE
2828
PUBLISH
SUBSCRIPTION REQUEST
SUBSCRIPTION REDIRECT
SUBSCRIBER
PUBLISHER
BROKER
CONFIGURATOR
TAD
2810

Fig. 27

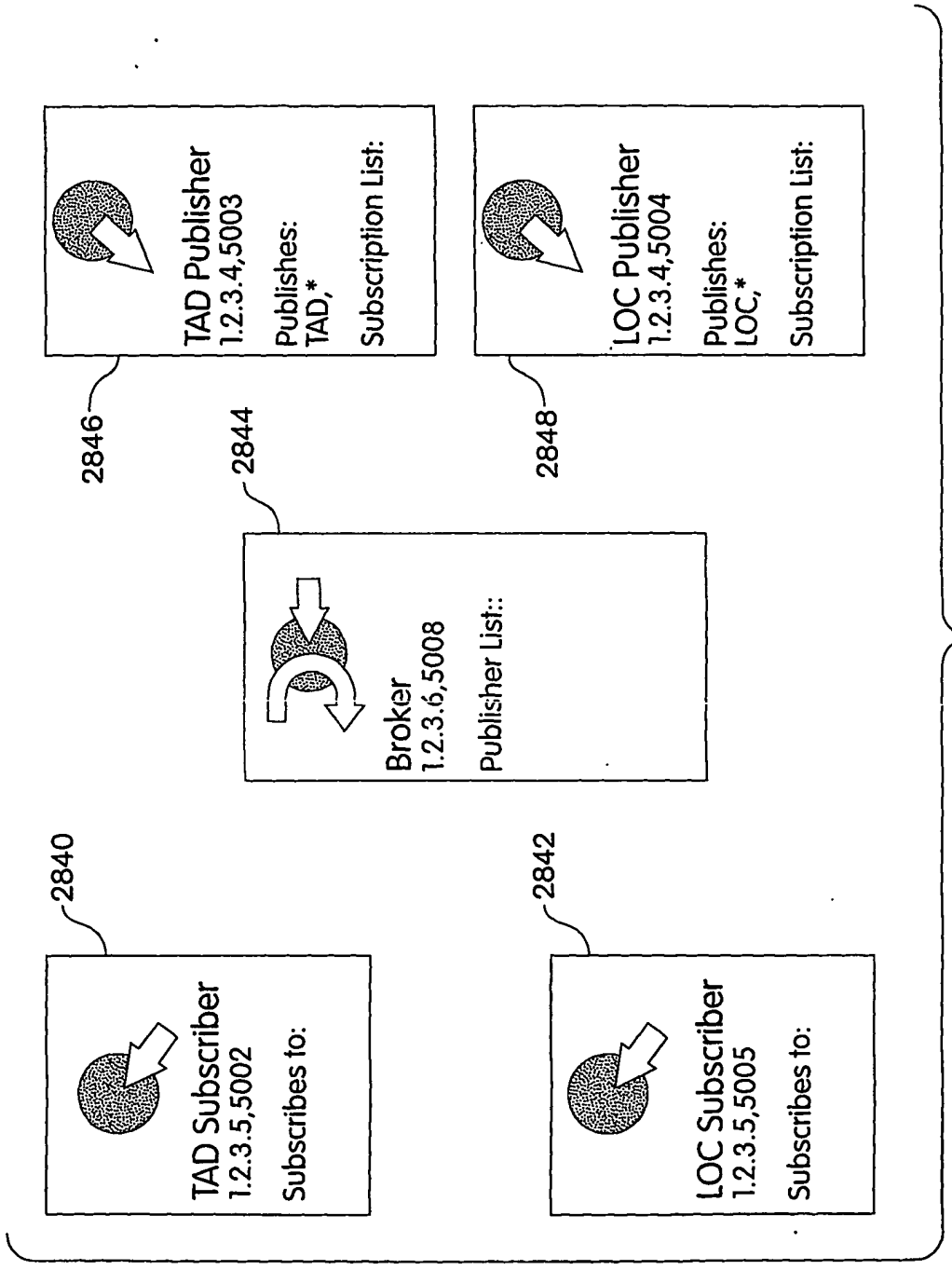

TAD Publisher
1.2.3.4,5003
Publishes:
TAD,*
Subscription List:
LOC Publisher
1.2.3.4,5004
Publishes:
LOC,*
Subscription List:
2846
2844
2848
Broker
1.2.3.6,5008
Publisher List::
2840
2842
TAD Subscriber
1.2.3.5,5002
Subscribes to:
LOC Subscriber
1.2.3.5,5005
Subscribes to:

Fig. 28

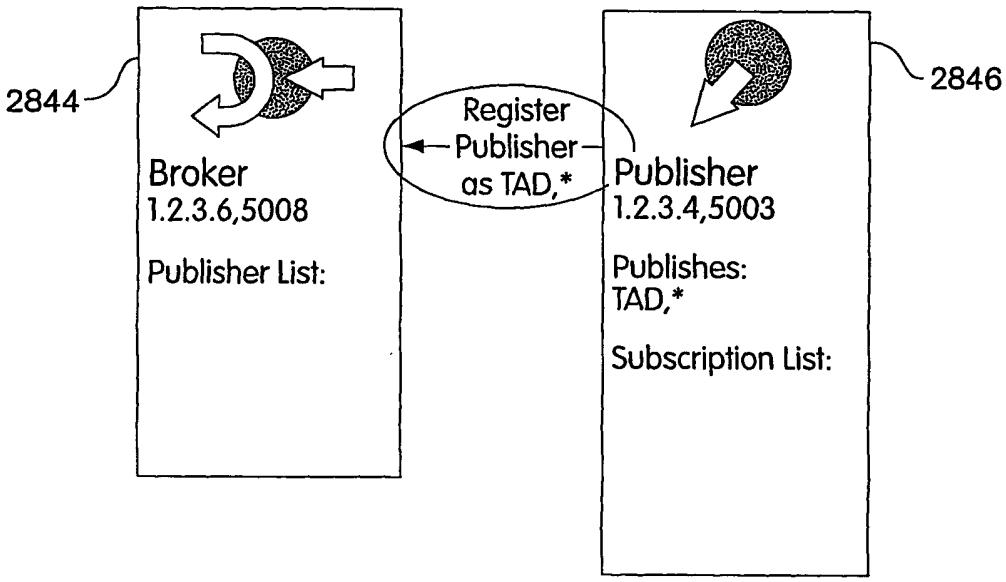

2844
Broker
1.2.3.6,5008
Publisher List:
Register
Publisher
as TAD,*
Publisher
1.2.3.4,5003
Publishes:
TAD,*
Subscription List:
2846

Fig. 29

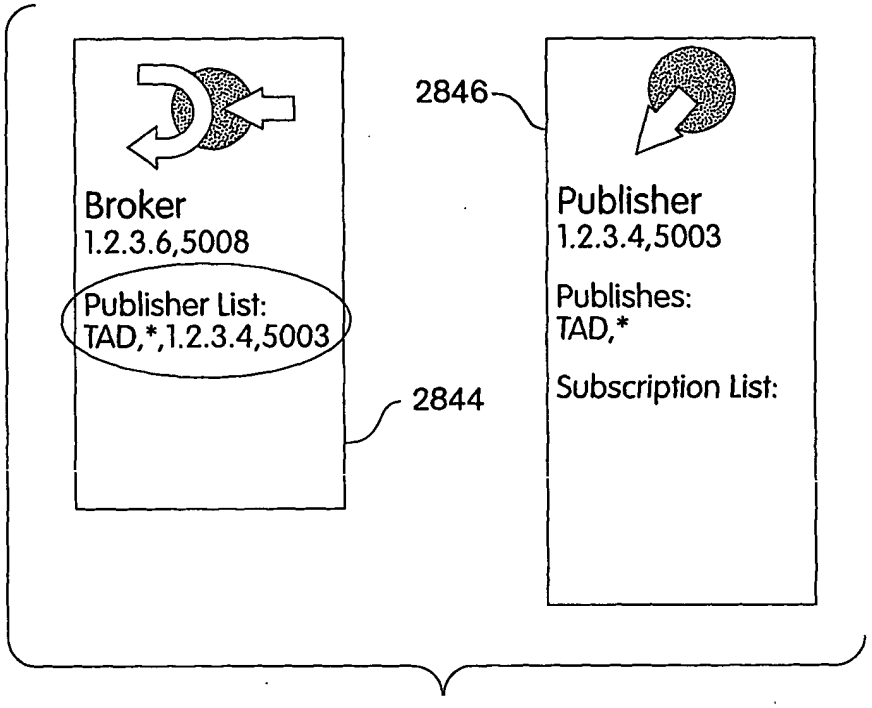

Broker
1.2.3.6,5008
Publisher List:
TAD,*,1.2.3.4,5003
2844
2846
Publisher
1.2.3.4,5003
Publishes:
TAD,*
Subscription List:

Fig. 30

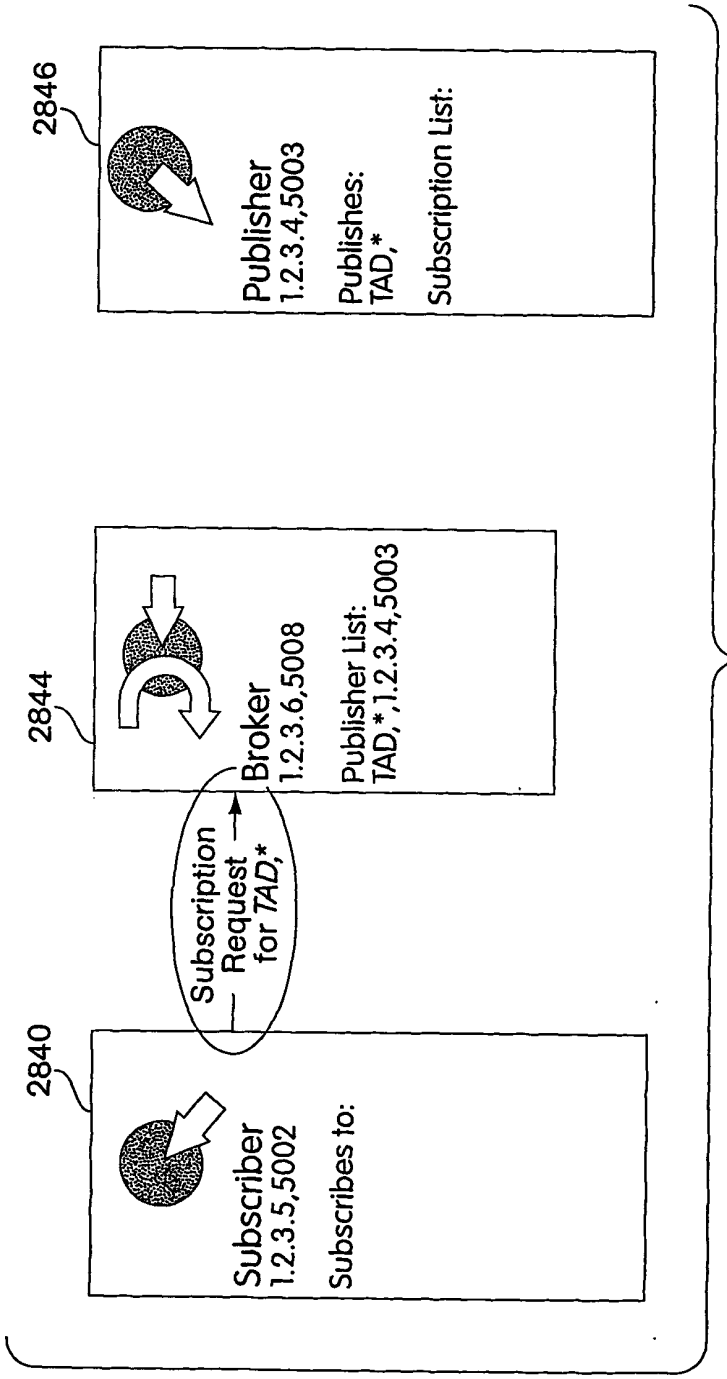

2846
Publisher
1.2.3.4,5003
Publishes:
TAD,*
Subscription List:
2844
Broker
1.2.3.6,5008
Publisher List:
TAD,*,1.2.3.4,5003
Subscription
Request
for TAD,*
2840
Subscriber
1.2.3.5,5002
Subscribes to:

Fig. 31

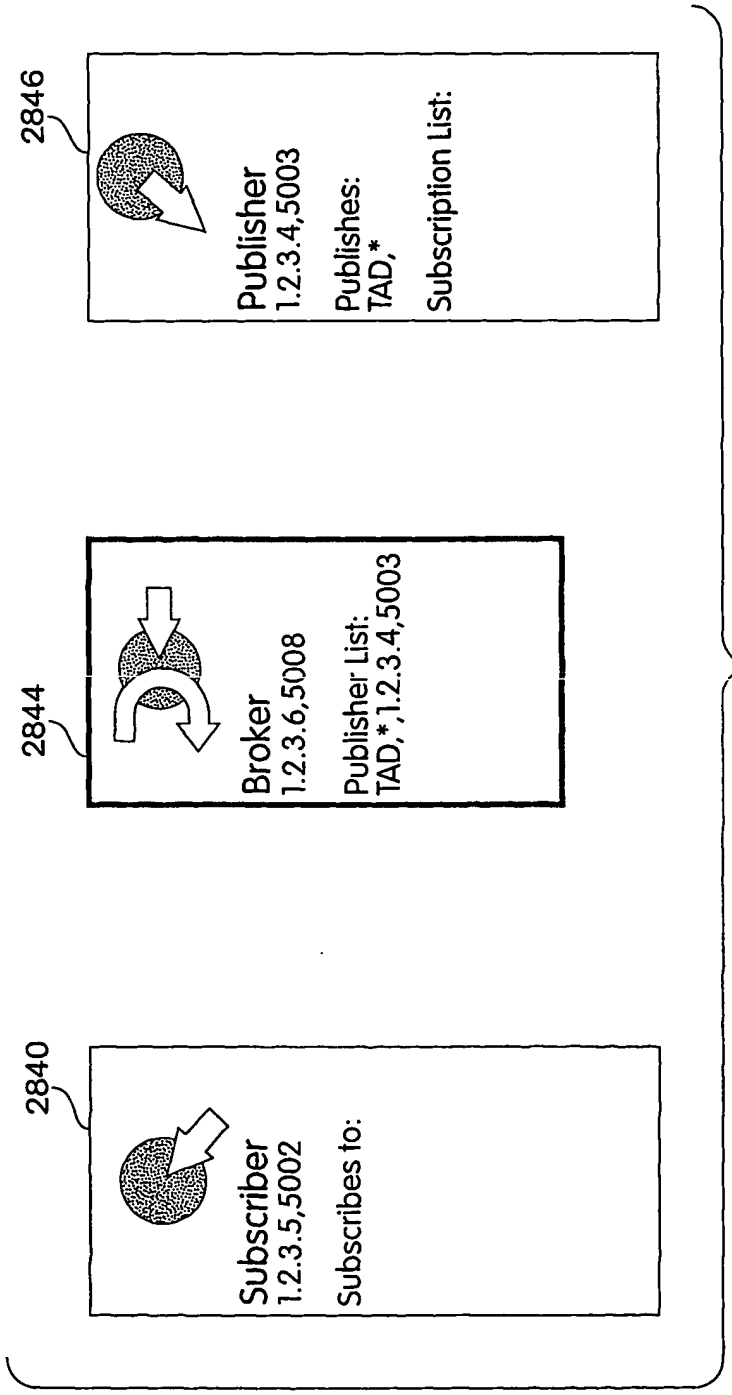

2846
Publisher
1.2.3.4,5003
Publishes:
TAD,*
Subscription List:
2844
Broker
1.2.3.6,5008
Publisher List:
TAD,*,1.2.3.4,5003
2840
Subscriber
1.2.3.5,5002
Subscribes to:

Fig. 32

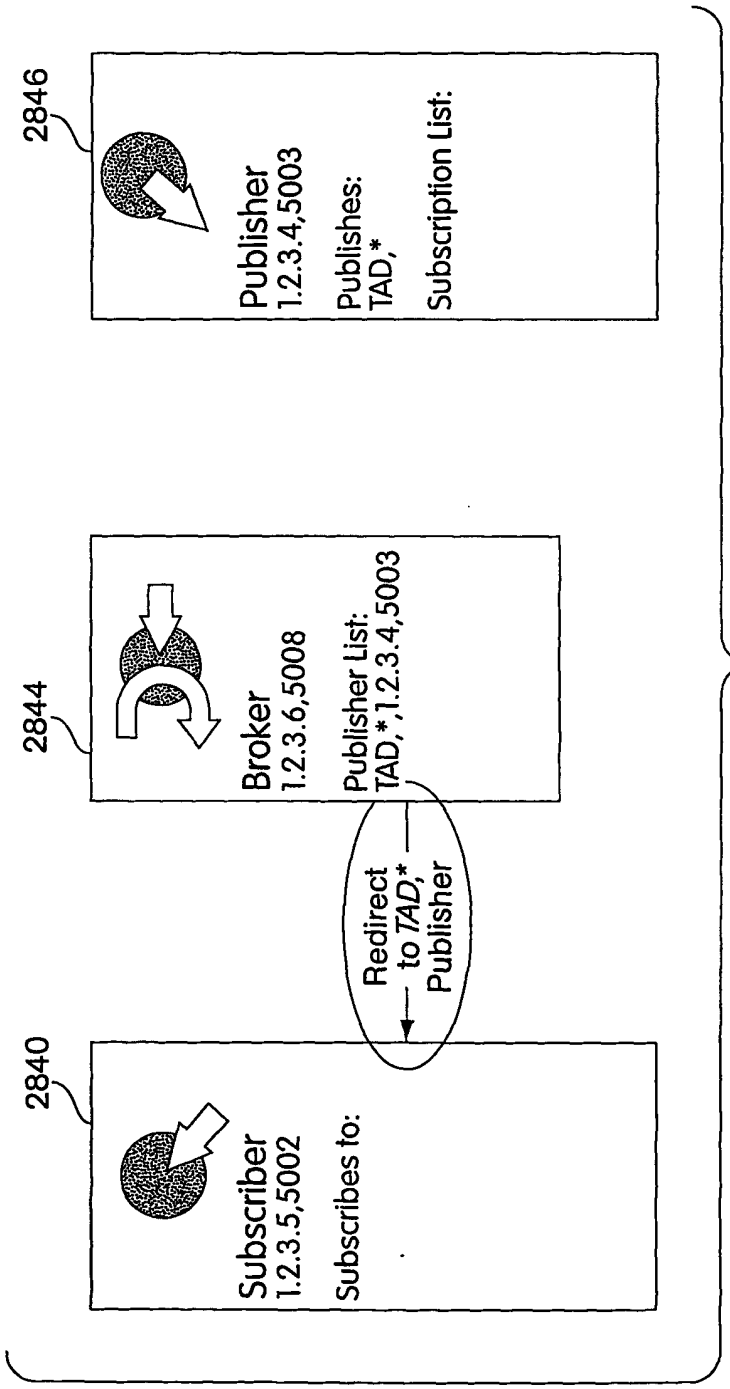

2846
Publisher
1.2.3.4,5003
Publishes:
TAD,*
Subscription List:
2844
Broker
1.2.3.6,5008
Publisher List:
TAD,*,1.2.3.4,5003
Redirect
to TAD,*
Publisher
2840
Subscriber
1.2.3.5,5002
Subscribes to:

Fig. 33

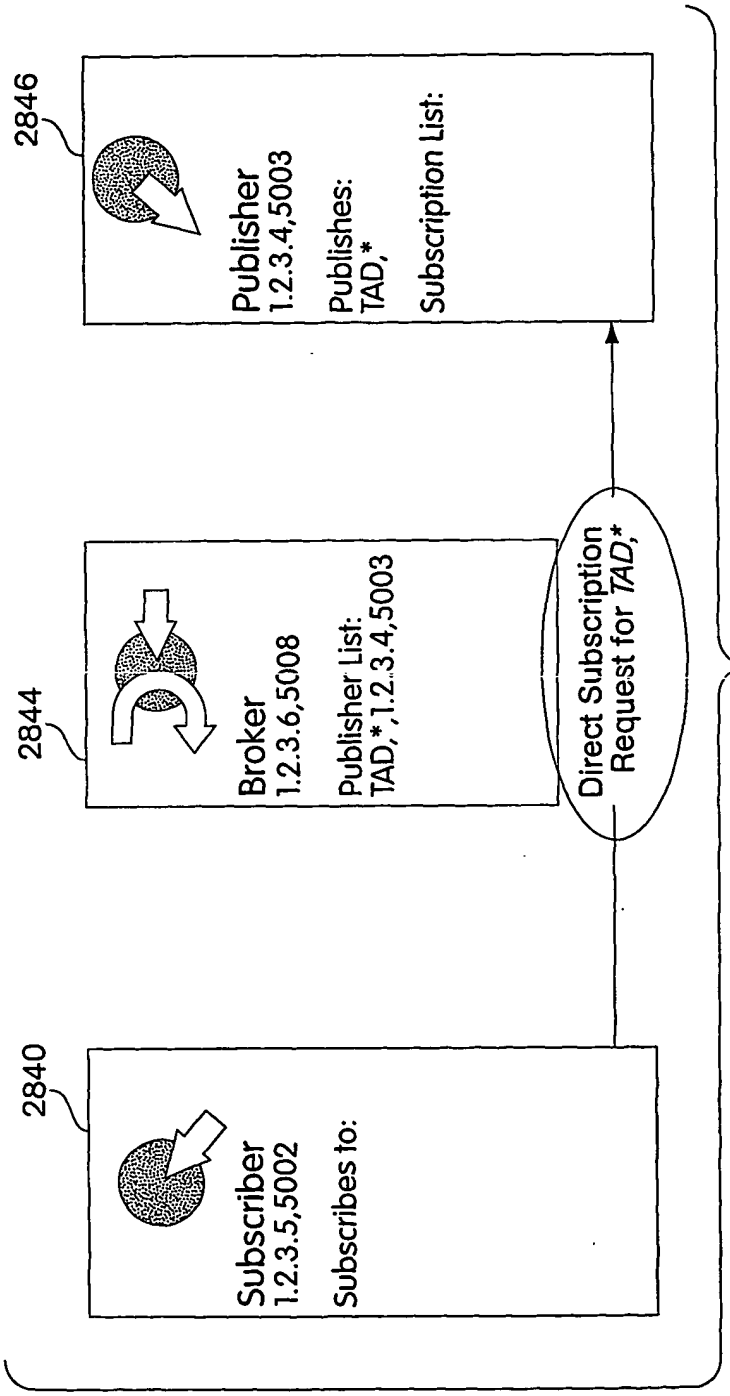

2846
Publisher
1.2.3.4,5003
Publishes:
TAD,*
Subscription List:
2844
Broker
1.2.3.6,5008
Publisher List:
TAD,*,1.2.3.4,5003
Direct Subscription
Request for TAD,*
2840
Subscriber
1.2.3.5,5002
Subscribes to:

Fig. 34

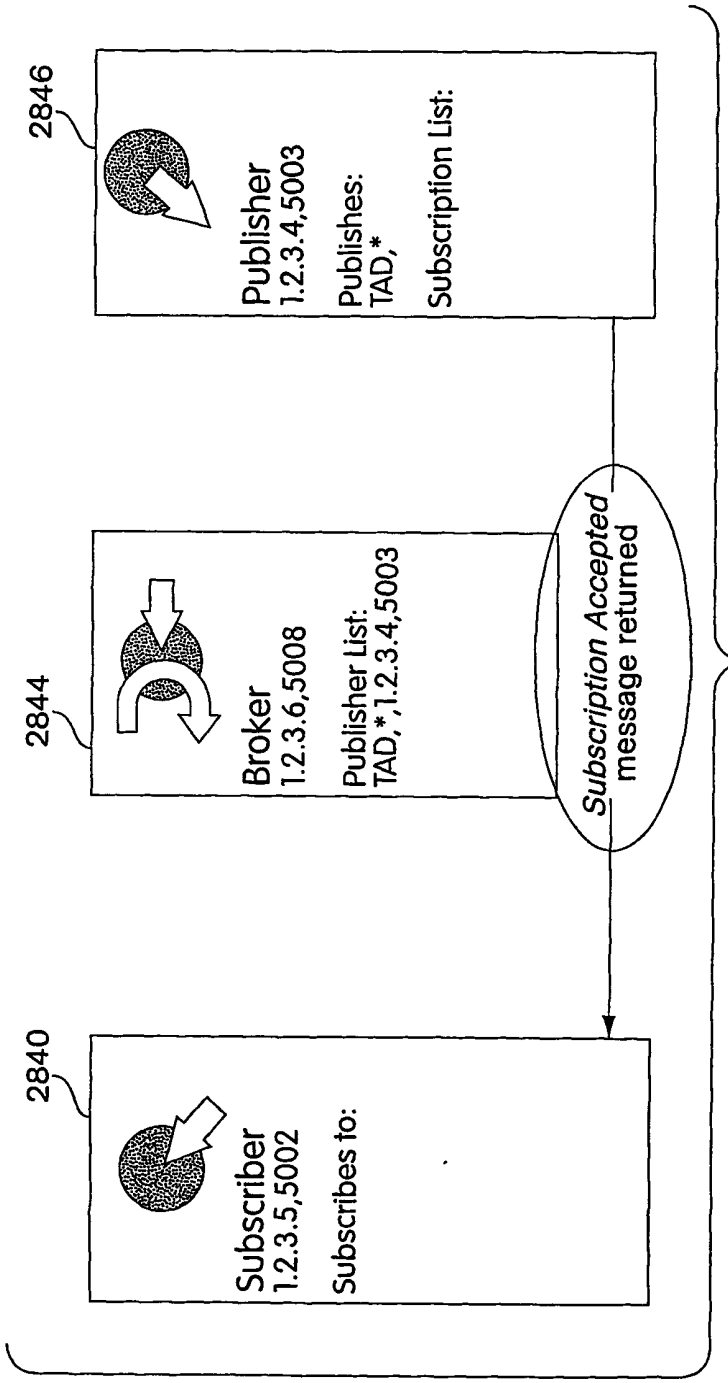

2846
Publisher
1.2.3.4,5003
Publishes:
TAD,*
Subscription List:
2844
Broker
1.2.3.6,5008
Publisher List:
TAD,*,1.2.3.4,5003
Subscription Accepted
message returned
2840
Subscriber
1.2.3.5,5002
Subscribes to:

Fig. 35

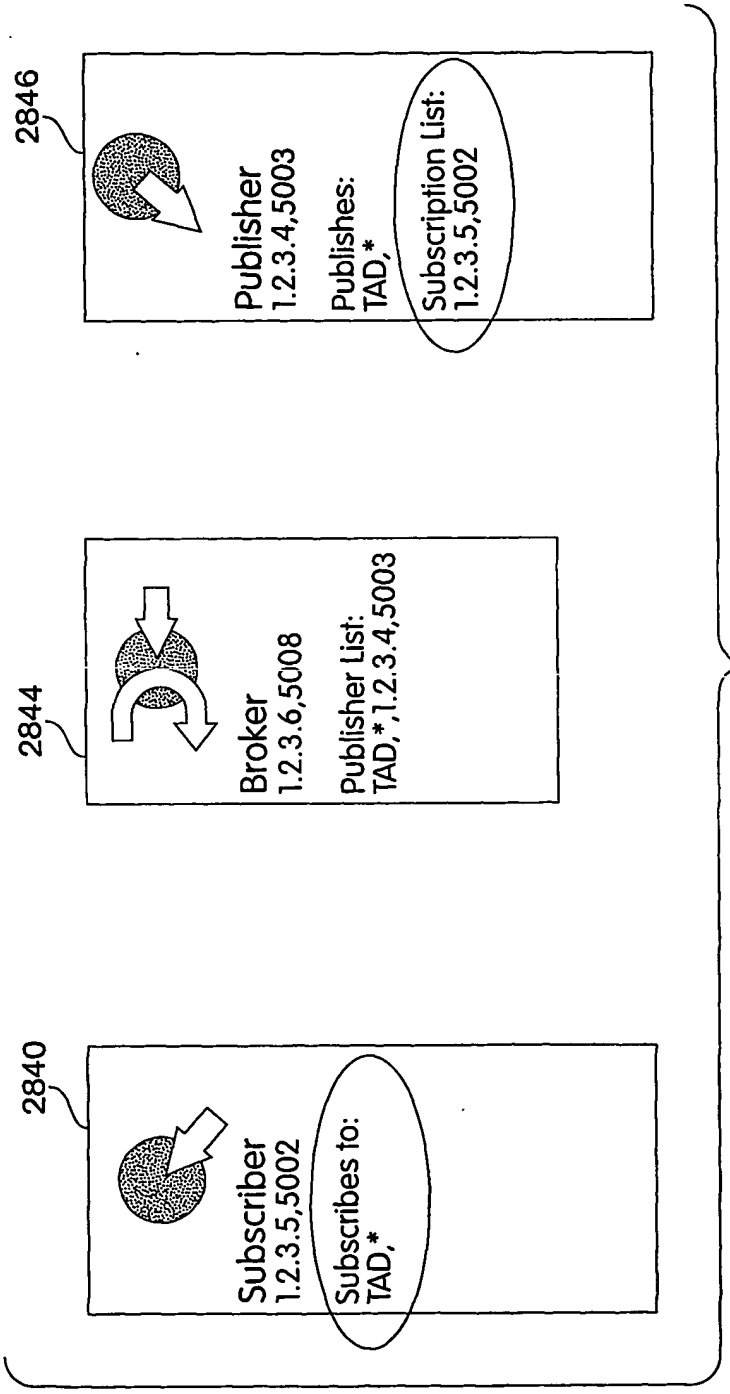

2846
Publisher
1.2.3.4,5003
Publishes:
TAD,*
Subscription List:
1.2.3.5,5002
2844
Broker
1.2.3.6,5008
Publisher List:
TAD,*,1.2.3.4,5003
2840
Subscriber
1.2.3.5,5002
Subscribes to:
TAD,*

Fig. 36

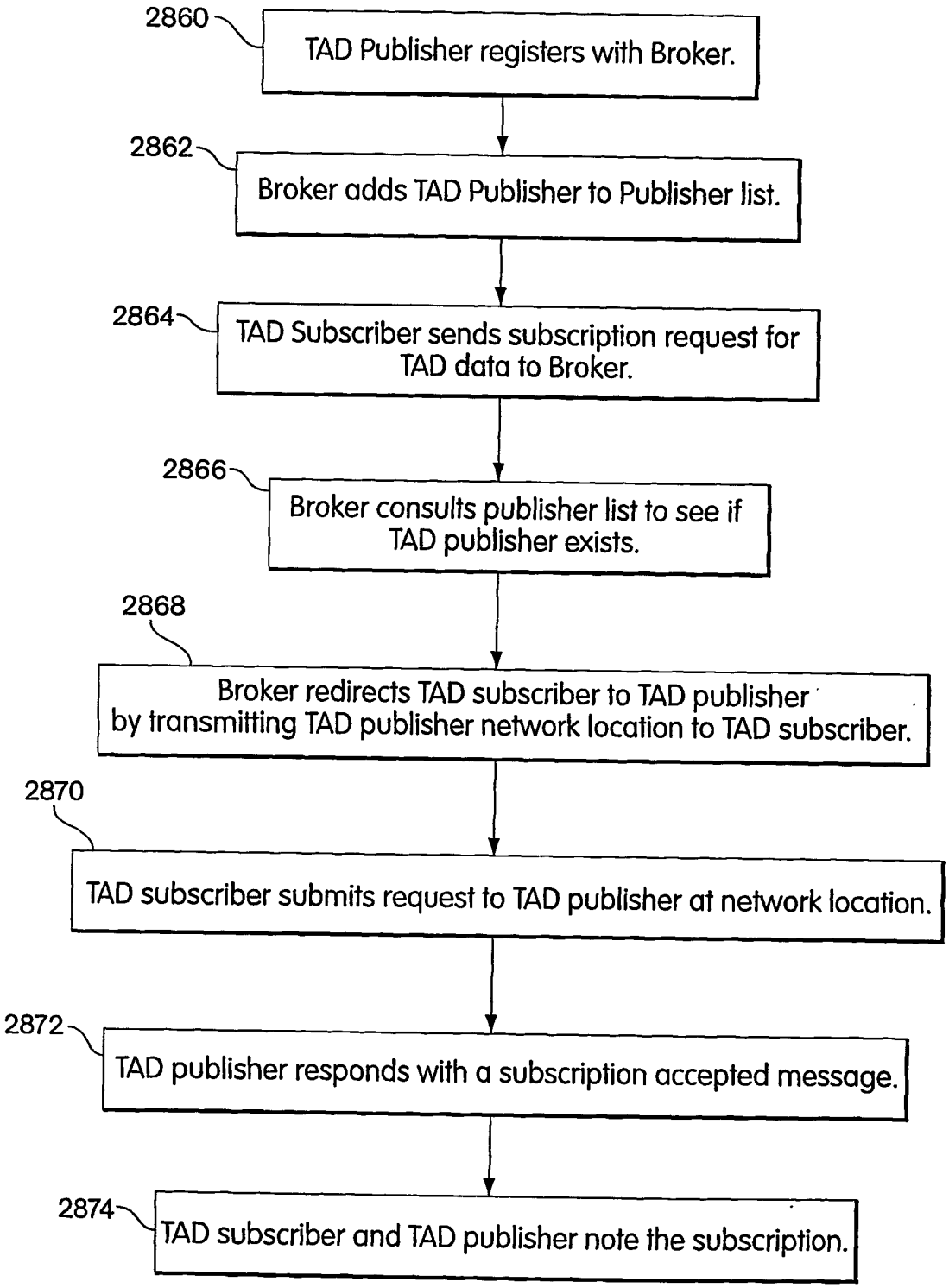

2860
TAD Publisher registers with Broker.
2862
Broker adds TAD Publisher to Publisher list.
2864
TAD Subscriber sends subscription request for TAD data to Broker.
2866
Broker consults publisher list to see if TAD publisher exists.
2868
Broker redirects TAD subscriber to TAD publisher by transmitting TAD publisher network location to TAD subscriber.
2870
TAD subscriber submits request to TAD publisher at network location.
2872
TAD publisher responds with a subscription accepted message.
2874
TAD subscriber and TAD publisher note the subscription.

Fig. 37

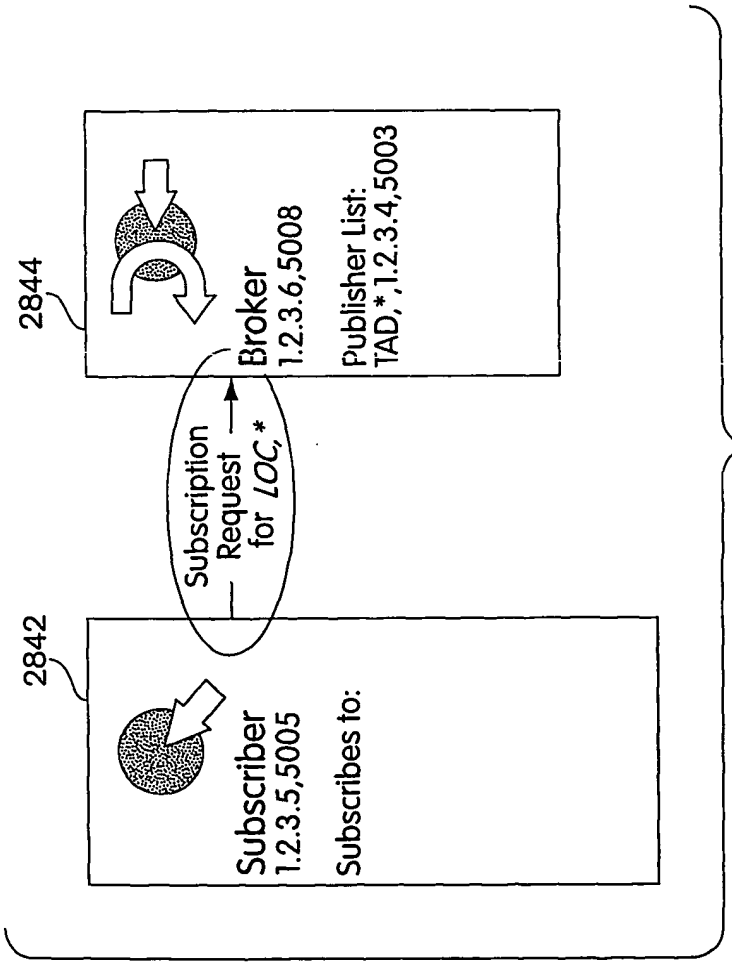

2844
Broker
1.2.3.6,5008
Publisher List:
TAD,*,1.2.3.4,5003
Subscription
Request
for LOC,*
2842
Subscriber
1.2.3.5,5005
Subscribes to:

Fig. 38

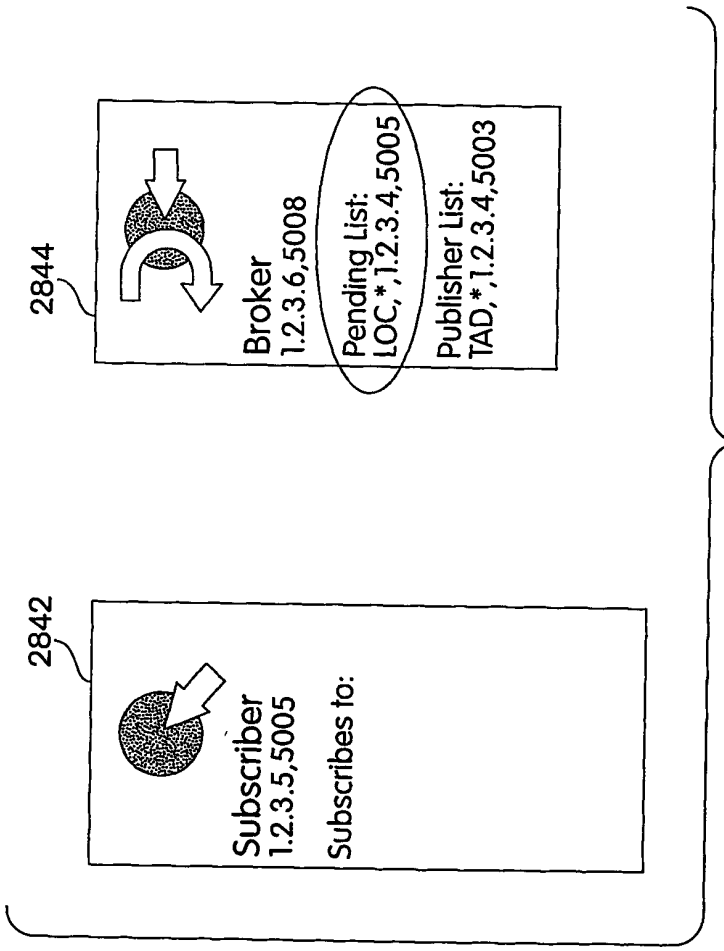

2842
Subscriber
1.2.3.5,5005
Subscribes to:
2844
Broker
1.2.3.6,5008
Pending List:
LOC,*,1.2.3.4,5005
Publisher List:
TAD,*,1.2.3.4,5003

Fig. 39

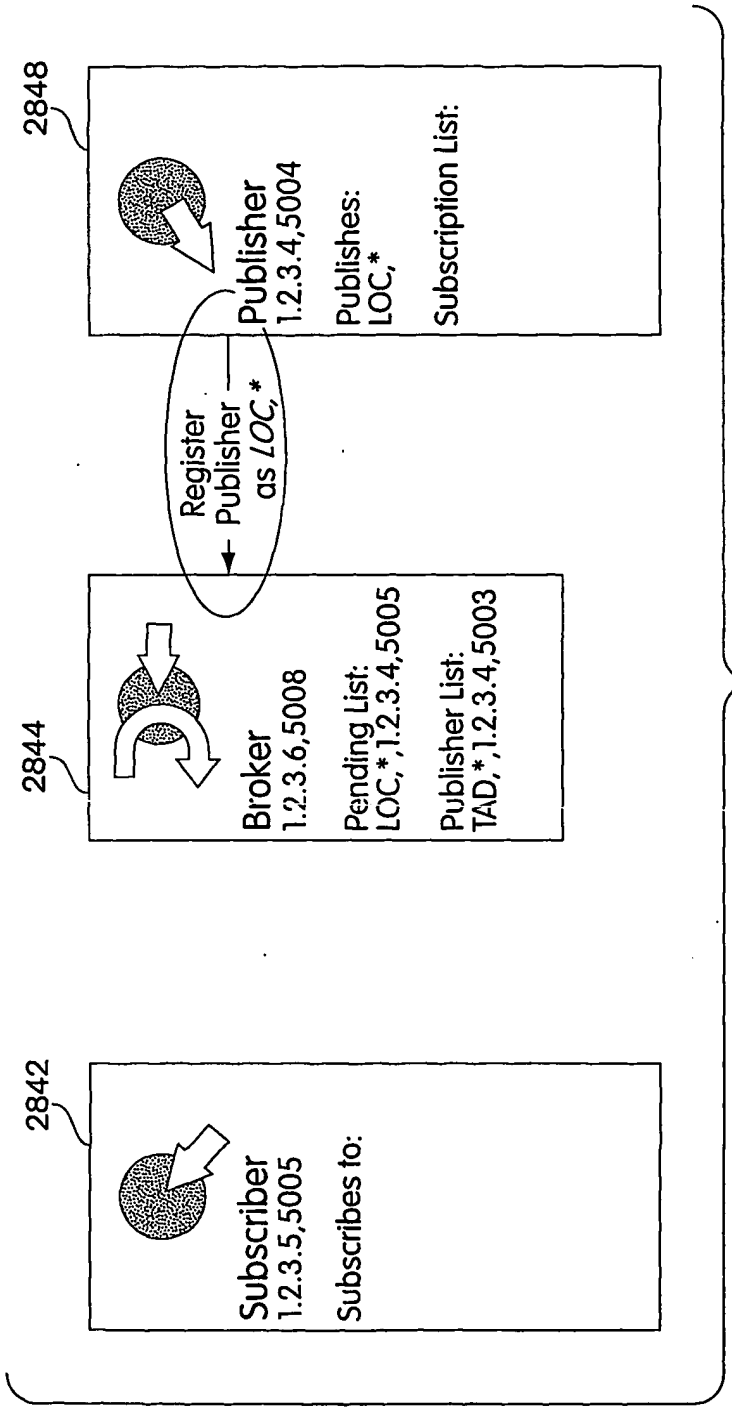

2848
Publisher
1.2.3.4,5004
Publishes:
LOC,*
Subscription List:
Register
Publisher
as LOC,*
2844
Broker
1.2.3.6,5008
Pending List:
LOC,*,1.2.3.4,5005
Publisher List:
TAD,*,1.2.3.4,5003
2842
Subscriber
1.2.3.5,5005
Subscribes to:

Fig. 40

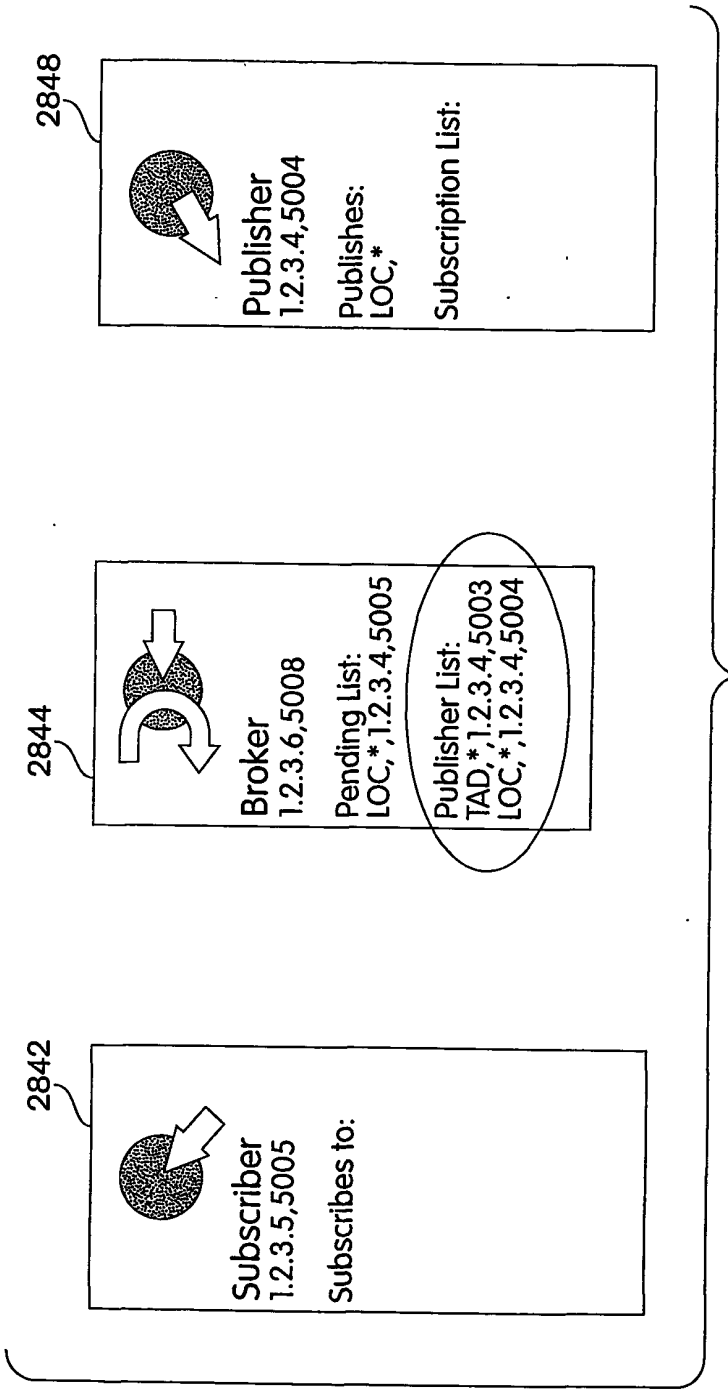

2848
Publisher
1.2.3.4,5004
Publishes:
LOC,*
Subscription List:
2844
Broker
1.2.3.6,5008
Pending List:
LOC,*,1.2.3.4,5005
Publisher List:
TAD,*,1.2.3.4,5003
LOC,*,1.2.3.4,5004
2842
Subscriber
1.2.3.5,5005
Subscribes to:

Fig. 41

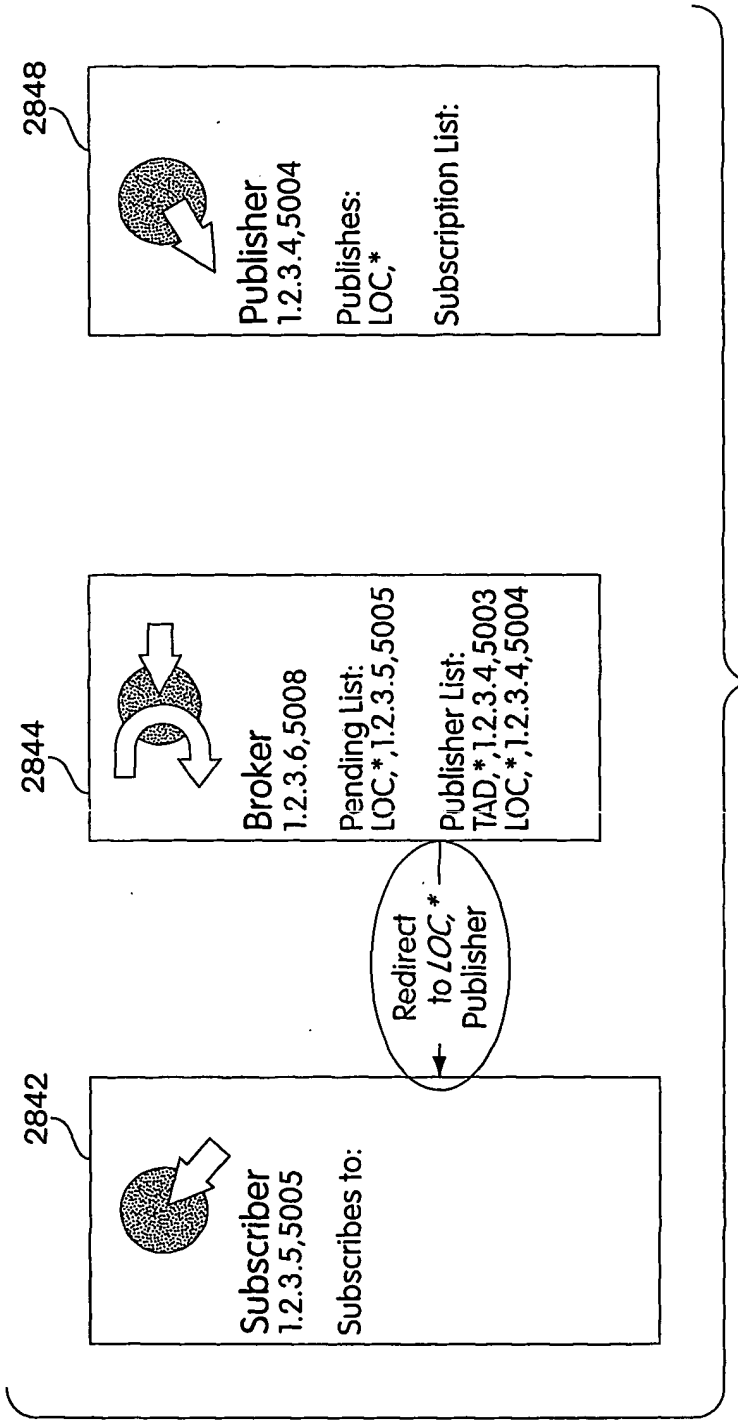

2848
Publisher
1.2.3.4,5004
Publishes:
LOC,*
Subscription List:
2844
Broker
1.2.3.6,5008
Pending List:
LOC,*,1.2.3.5,5005
Publisher List:
TAD,*,1.2.3.4,5003
LOC,*,1.2.3.4,5004
Redirect
to LOC,*
Publisher
2842
Subscriber
1.2.3.5,5005
Subscribes to:

Fig. 42

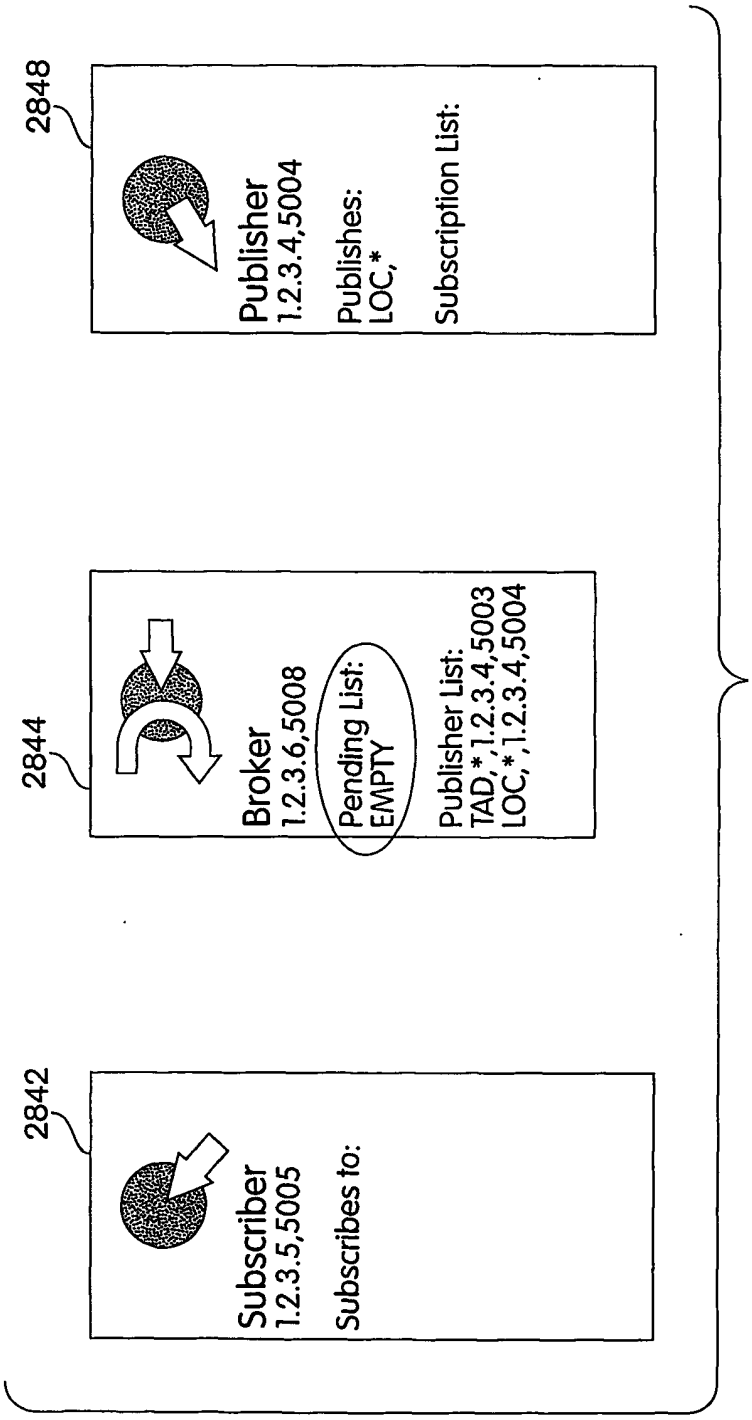

2842
Subscriber
1.2.3.5,5005
Subscribes to:
2844
Broker
1.2.3.6,5008
Pending List:
EMPTY
Publisher List:
TAD,*,1.2.3.4,5003
LOC,*,1.2.3.4,5004
2848
Publisher
1.2.3.4,5004
Publishes:
LOC,*
Subscription List:

Fig. 43

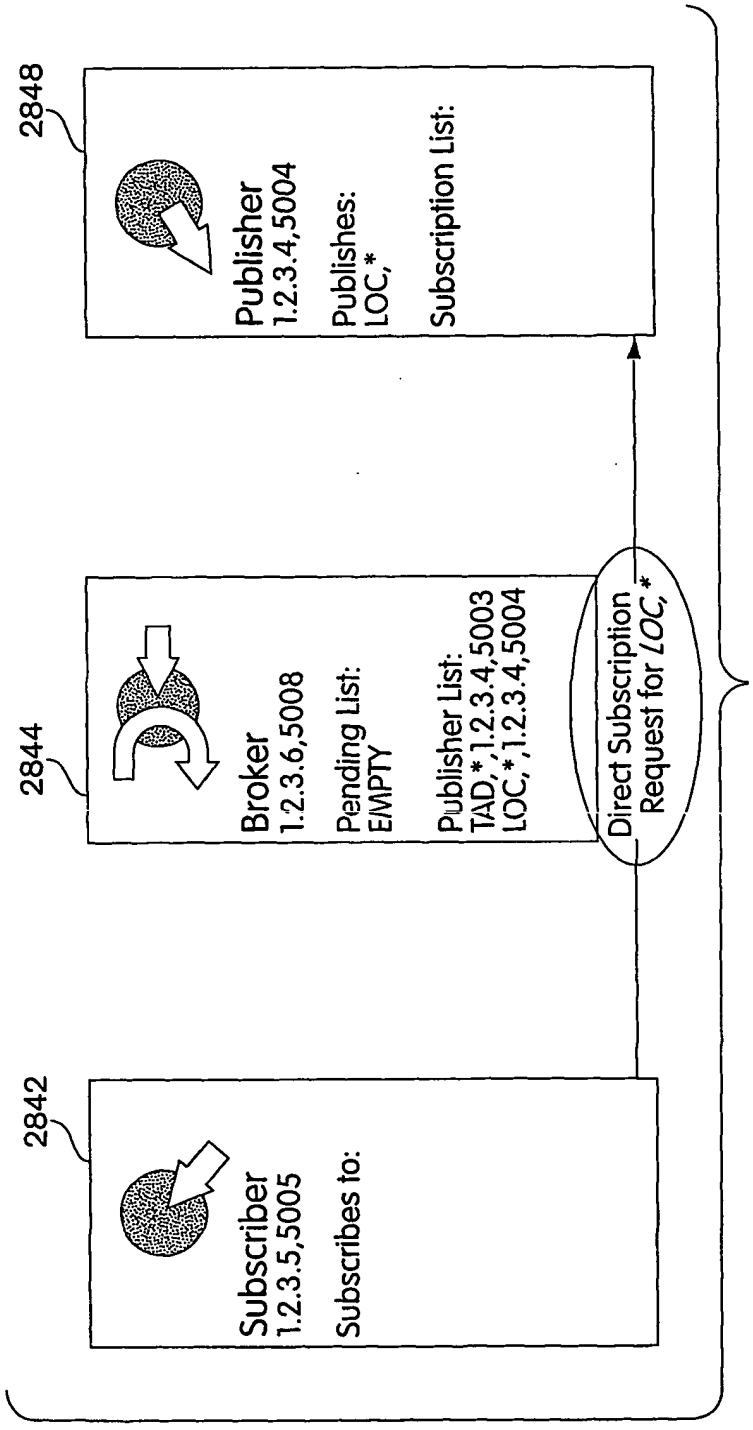

2848
Publisher
1.2.3.4,5004
Publishes:
LOC,*
Subscription List:
2844
Broker
1.2.3.6,5008
Pending List:
EMPTY
Publisher List:
TAD,*,1.2.3.4,5003
LOC,*,1.2.3.4,5004
Direct Subscription
Request for LOC,*
2842
Subscriber
1.2.3.5,5005
Subscribes to:

Fig. 44

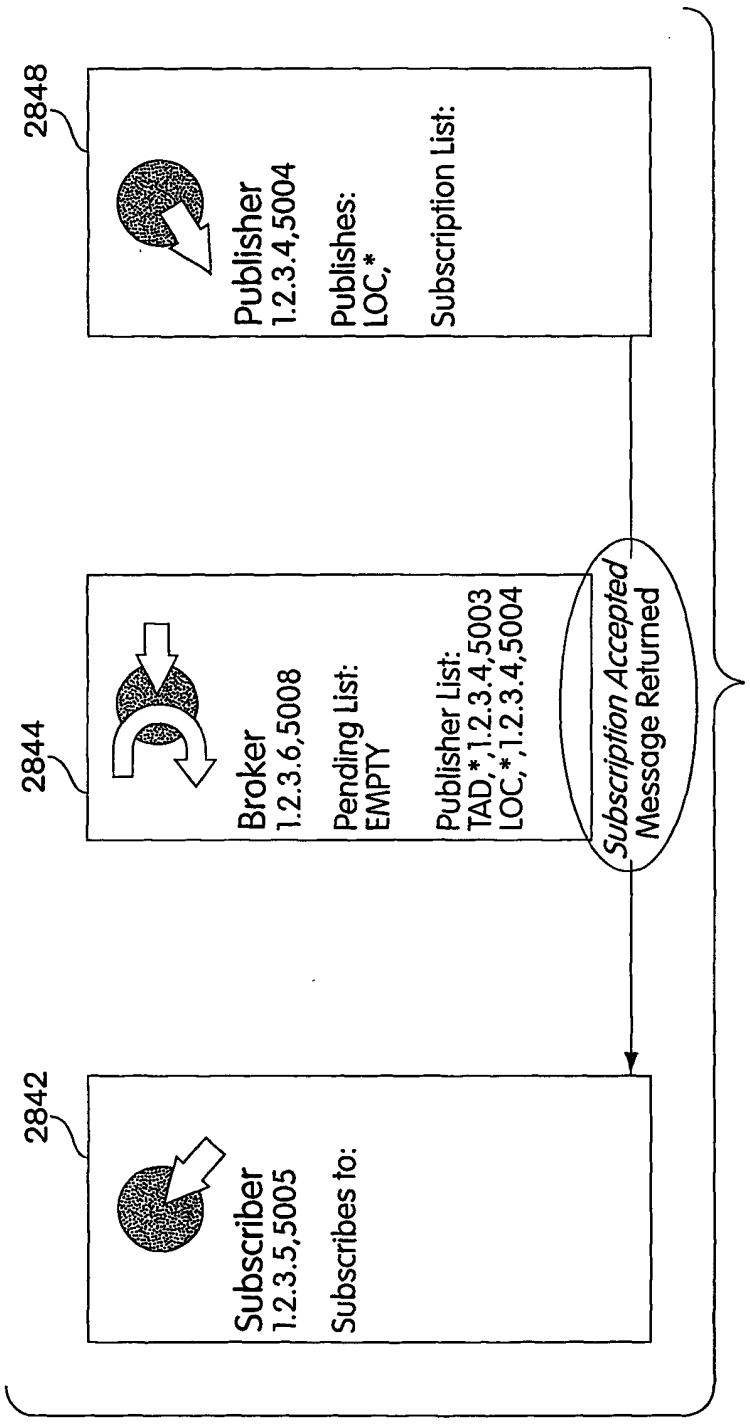

2848
Publisher
1.2.3.4,5004
Publishes:
LOC,*
Subscription List:
2844
Broker
1.2.3.6,5008
Pending List:
EMPTY
Publisher List:
TAD,*,1.2.3.4,5003
LOC,*,1.2.3.4,5004
Subscription Accepted
Message Returned
2842
Subscriber
1.2.3.5,5005
Subscribes to:

Fig. 45

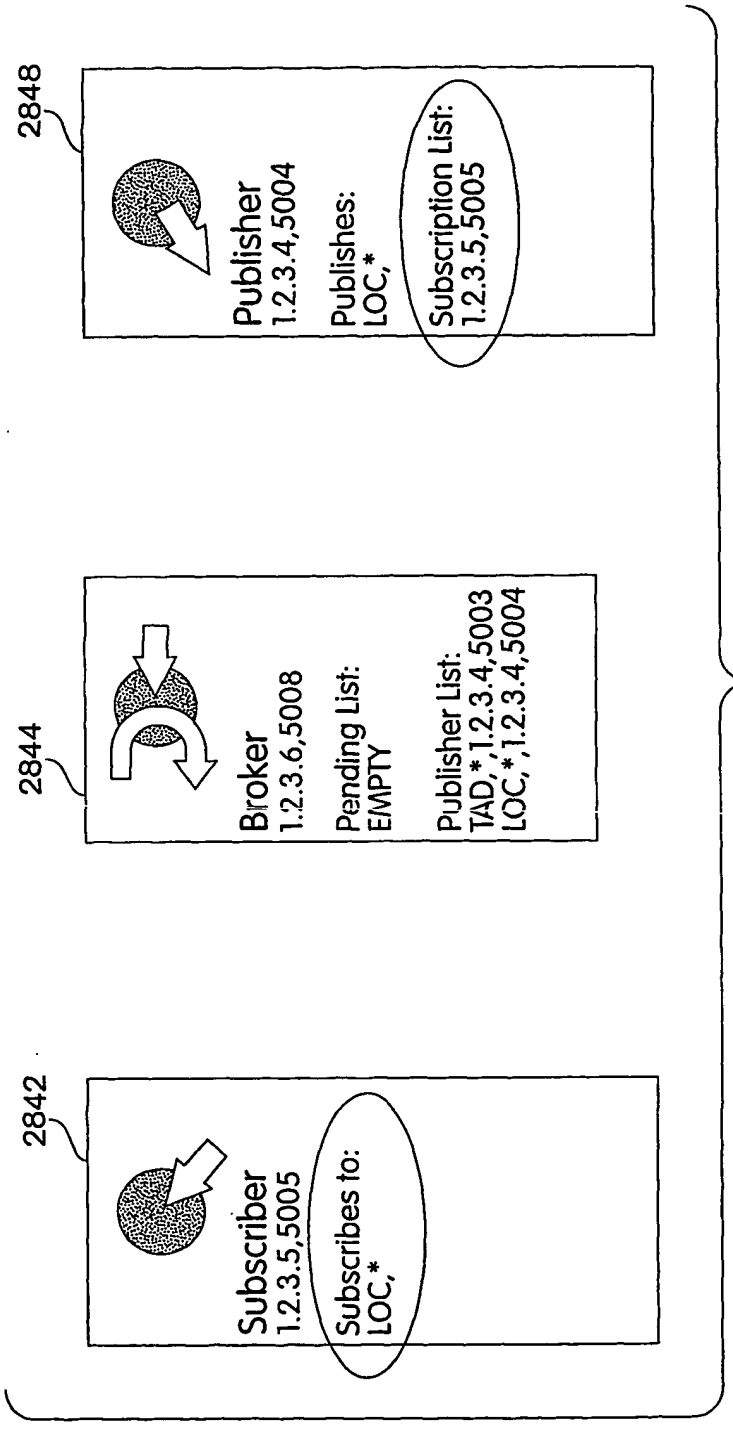

2842
Subscriber
1.2.3.5,5005
Subscribes to:
LOC,*
2844
Broker
1.2.3.6,5008
Pending List:
EMPTY
Publisher List:
TAD,*,1.2.3.4,5003
LOC,*,1.2.3.4,5004
2848
Publisher
1.2.3.4,5004
Publishes:
LOC,*
Subscription List:
1.2.3.5,5005

Fig. 46

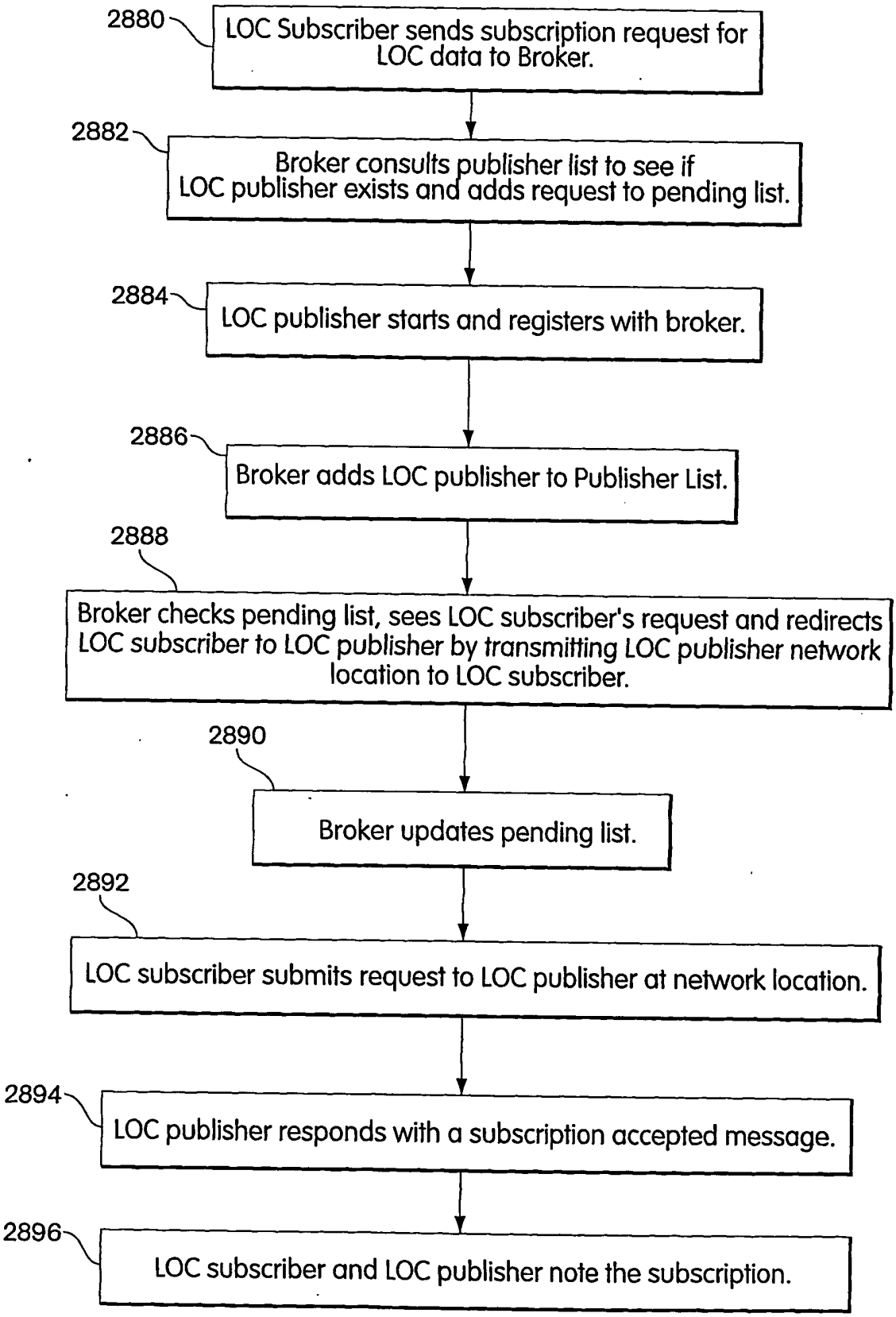

2880
LOC Subscriber sends subscription request for LOC data to Broker.
2882
Broker consults publisher list to see if LOC publisher exists and adds request to pending list.
2884
LOC publisher starts and registers with broker.
2886
Broker adds LOC publisher to Publisher List.
2888
Broker checks pending list, sees LOC subscriber's request and redirects LOC subscriber to LOC publisher by transmitting LOC publisher network location to LOC subscriber.
2890
Broker updates pending list.
2892
LOC subscriber submits request to LOC publisher at network location.
2894
LOC publisher responds with a subscription accepted message.
2896
LOC subscriber and LOC publisher note the subscription.

Fig. 47

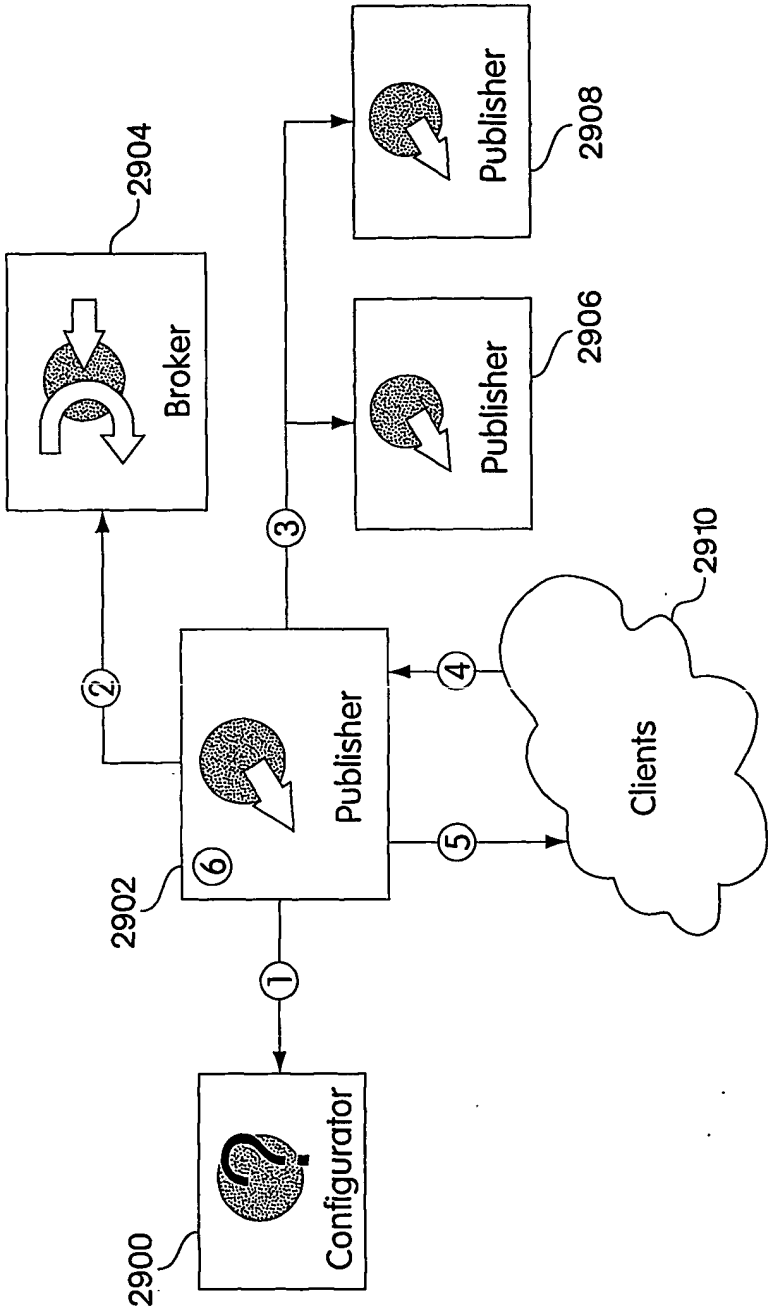

Configurator
2900
Publisher
2902
Broker
2904
Publisher
2906
Publisher
2908
Clients
2910
1
2
3
4
5
6

Fig. 48

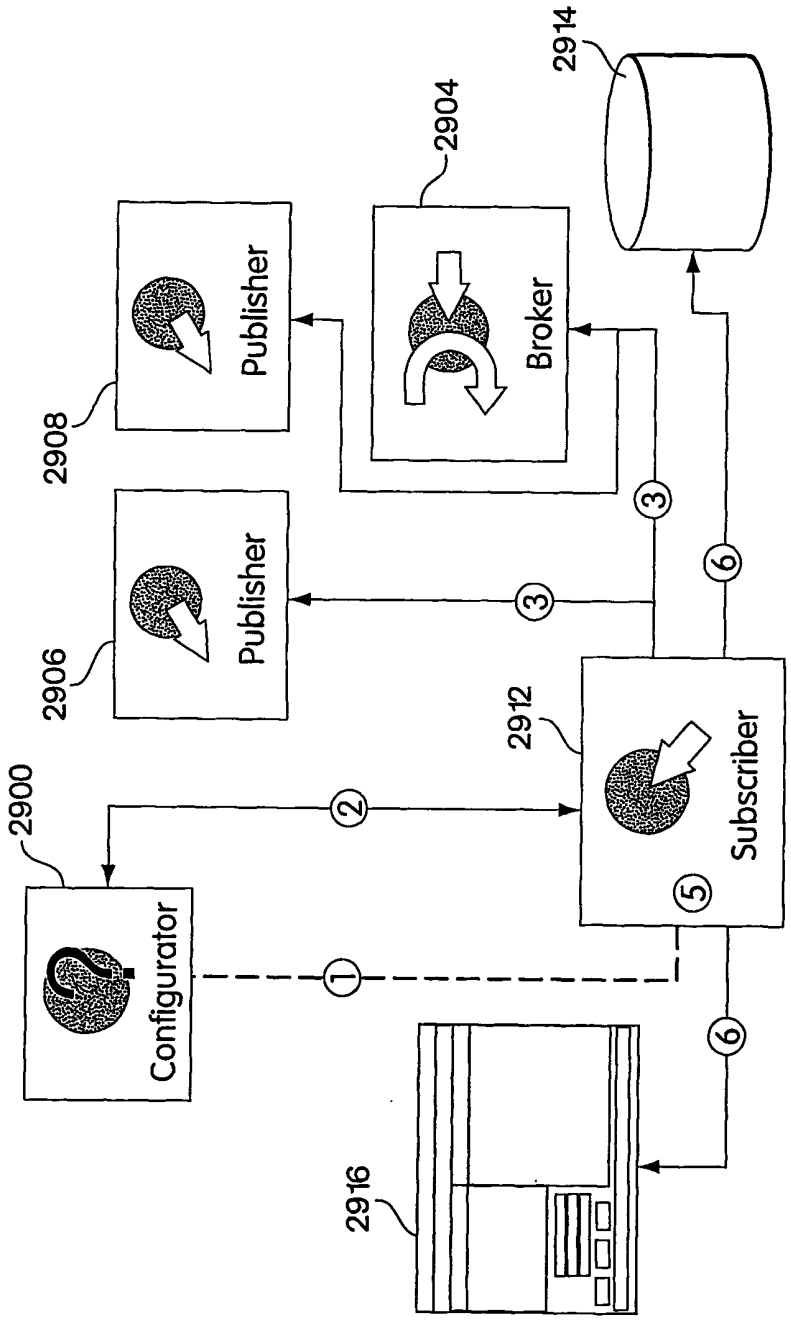

2914
2904
Broker
Publisher
2908
Publisher
2906
2912
Subscriber
2900
Configurator
2916
1
2
3
3
5
6
6

Fig. 49

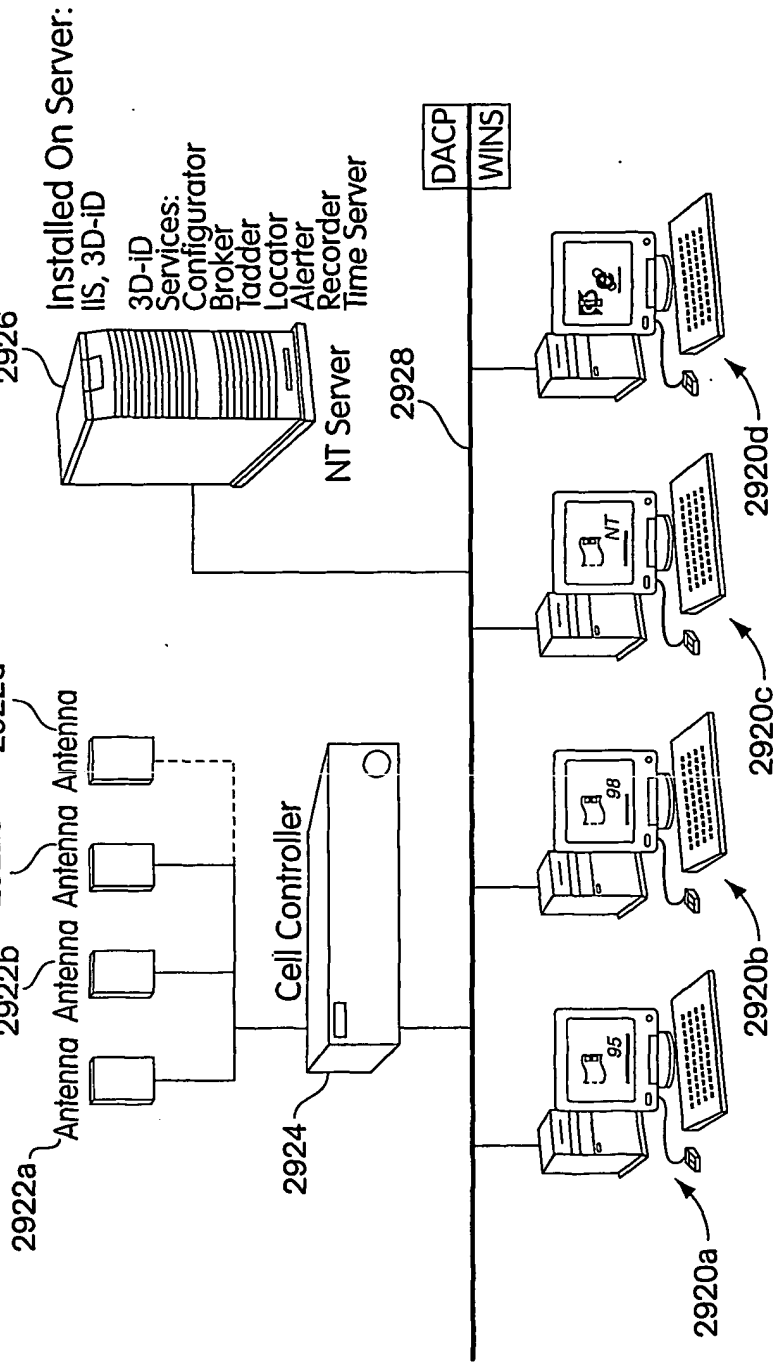

Installed On Server:
IIS, 3D-iD
3D-iD Services:
Configurator
Broker
Tadder
Locator
Alerter
Recorder
Time Server
NT Server
2926
2928
DACP
WINS
2920a
2920b
2920c
2920d
95
98
NT
Cell Controller
2924
2922a
2922b
2922c
2922d
Antenna
Antenna
Antenna
Antenna

Fig. 50

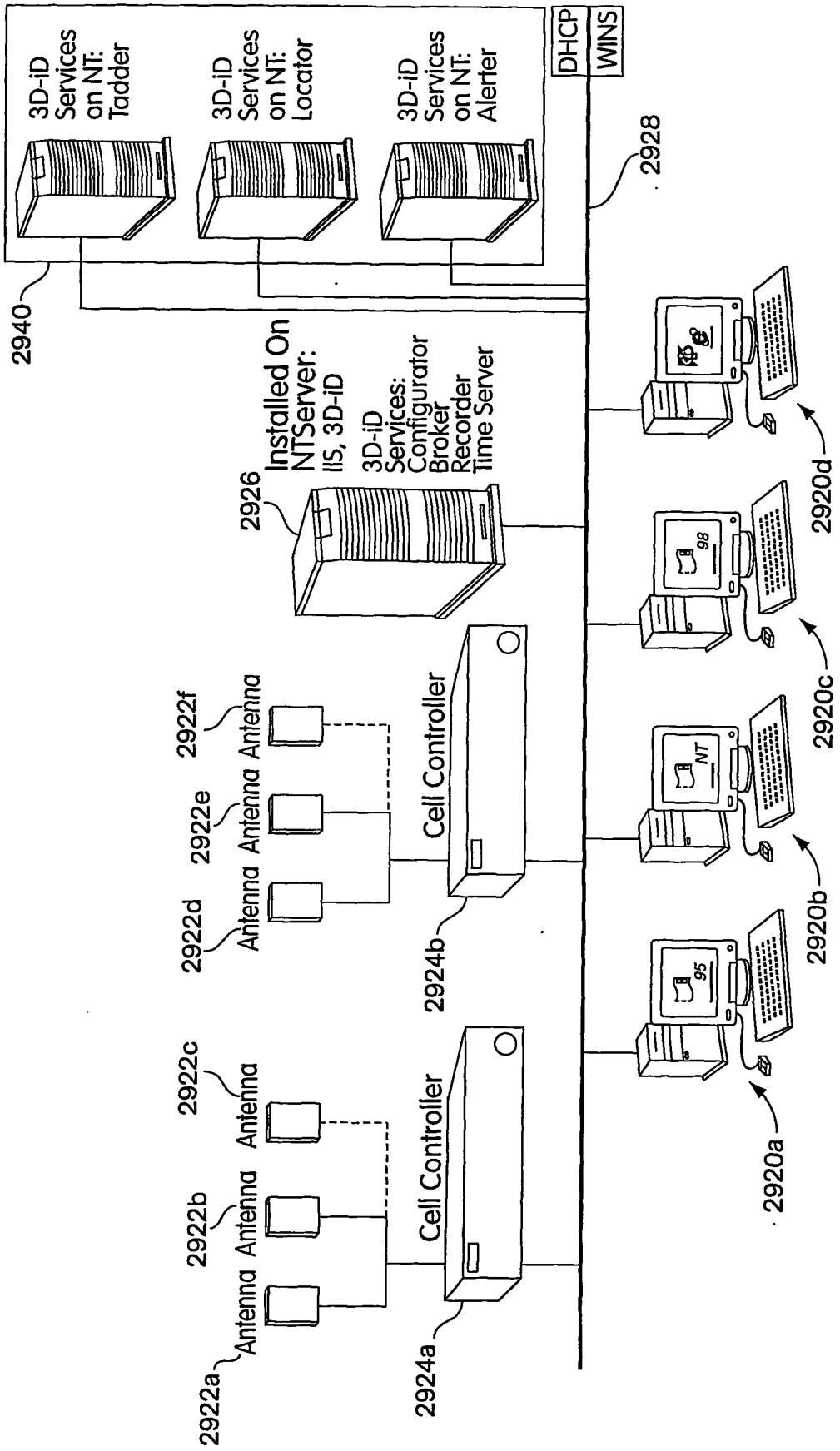

3D-iD Services on NT: Tadder
3D-iD Services on NT: Locator
3D-iD Services on NT: Alerter
DHCP
WINS
2928
2940
Installed On NTServer: IIS, 3D-iD
3D-iD Services: Configurator Broker Recorder Time Server
2926
2922f
Antenna
2922e
Antenna
2922d
Antenna
Cell Controller
2924b
2922c
Antenna
2922b
Antenna
Antenna
2922a
Cell Controller
2924a
2920d
2920c
2920b
2920a

Fig. 51

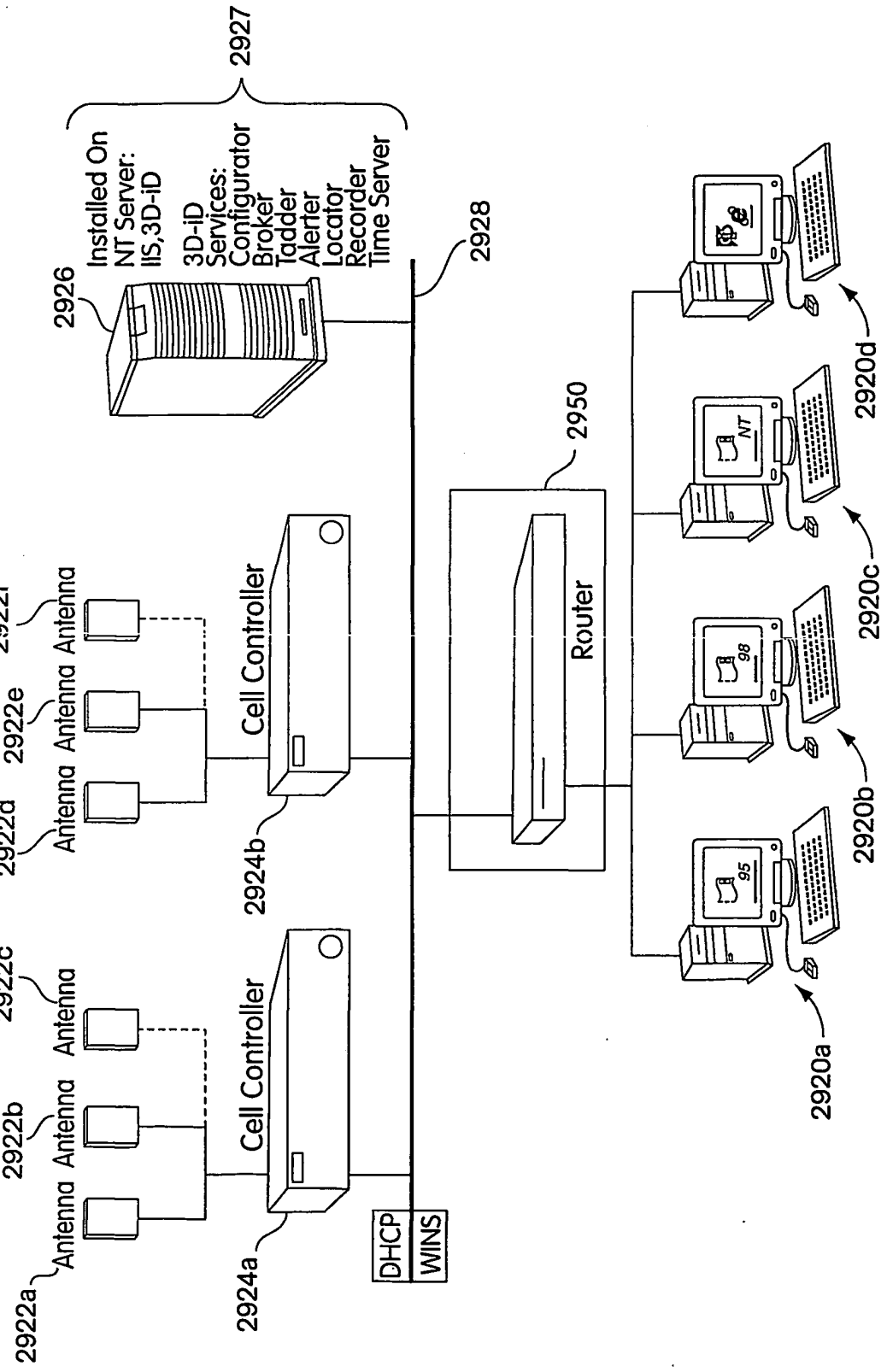

2927
Installed On
NT Server:
IIS,3D-iD
3D-iD
Services:
Configurator
Broker
Tadder
Alerter
Locator
Recorder
Time Server
2926
2928
2950
Router
2922f
Antenna
2922e
Antenna
2922d
Antenna
Cell Controller
2924b
2922c
Antenna
2922b
Antenna
2922a
Antenna
Cell Controller
2924a
DHCP
WINS
2920a
2920b
2920c
2920d

Fig. 52

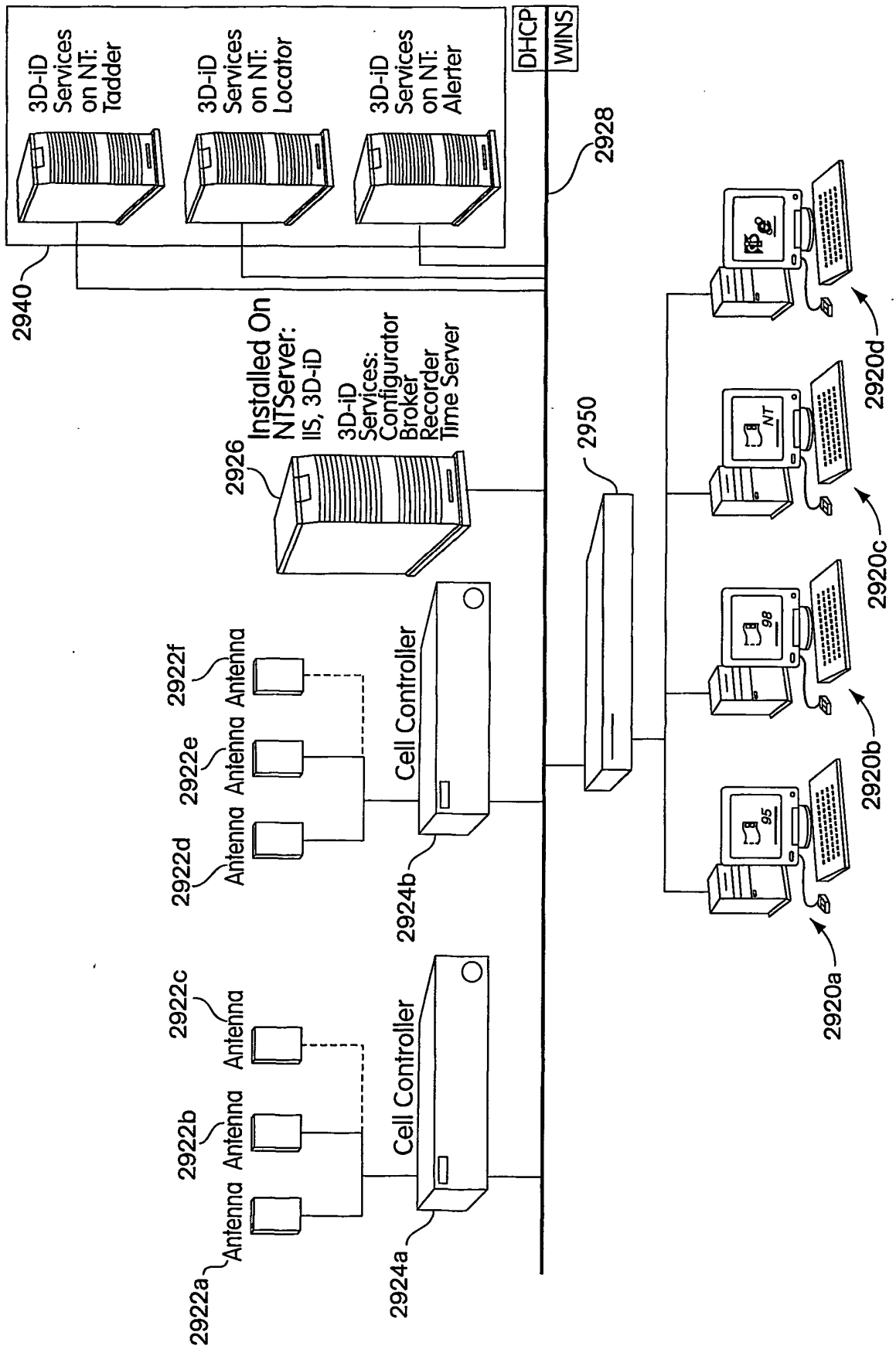

3D-iD Services on NT: Tadder
3D-iD Services on NT: Locator
3D-iD Services on NT: Alerter
DHCP
WINS
2928
2940
Installed On NTServer: IIS, 3D-iD
3D-iD Services: Configurator Broker Recorder Time Server
2926
2950
2922f
2922e
2922d
Antenna
Antenna
Antenna
Cell Controller
2924b
2922c
2922b
2922a
Antenna
Antenna
Antenna
Cell Controller
2924a
2920a
2920b
2920c
2920d
95
98
NT

Fig. 53

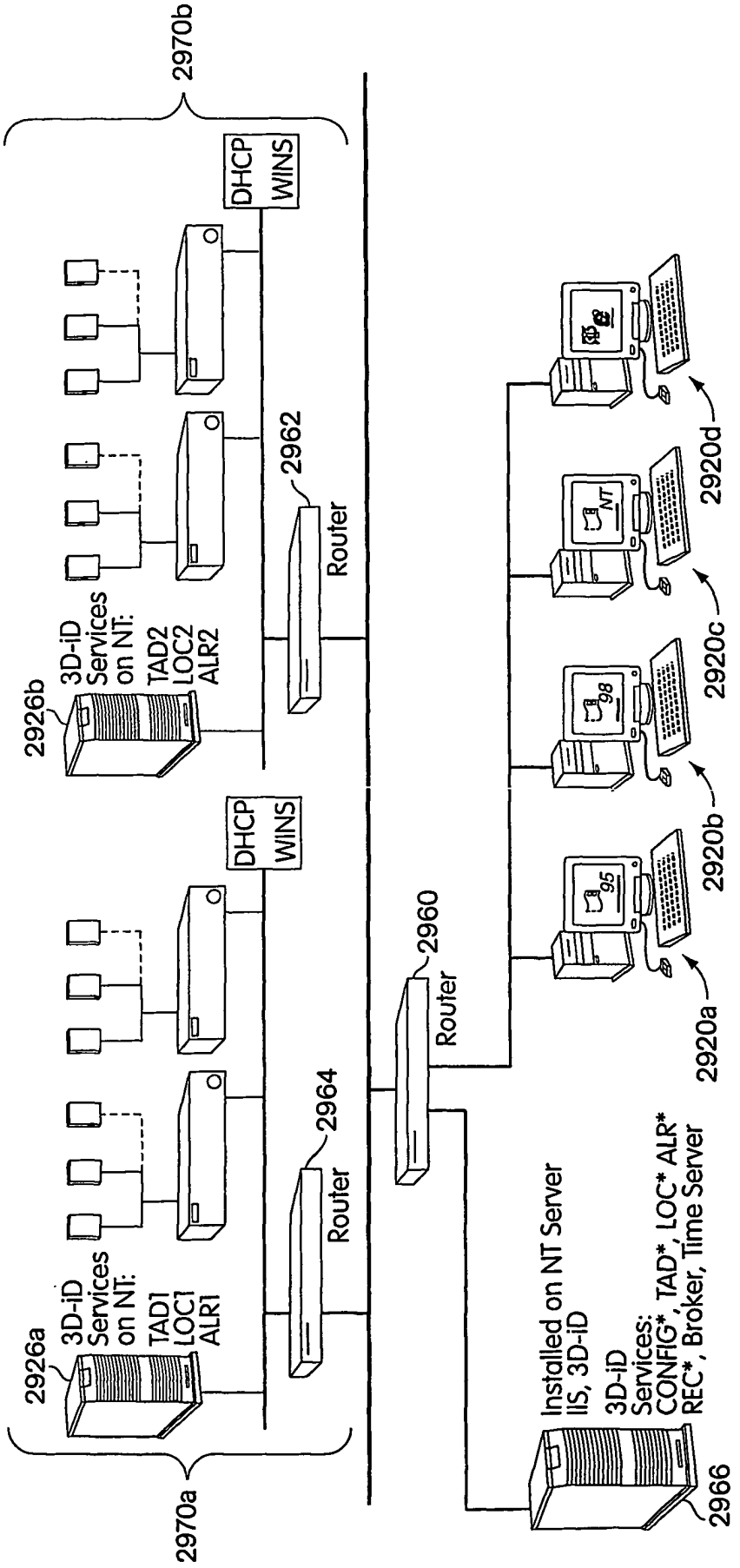

2970b
DHCP
WINS
2962
Router
2926b
3D-iD Services on NT:
TAD2
LOC2
ALR2
DHCP
WINS
2964
Router
2926a
3D-iD Services on NT:
TAD1
LOC1
ALR1
2970a
2960
Router
Installed on NT Server
IIS, 3D-iD
3D-iD Services:
CONFIG*, TAD*, LOC* ALR*
REC*, Broker, Time Server
2966
2920a
2920b
2920c
2920d

Fig. 54

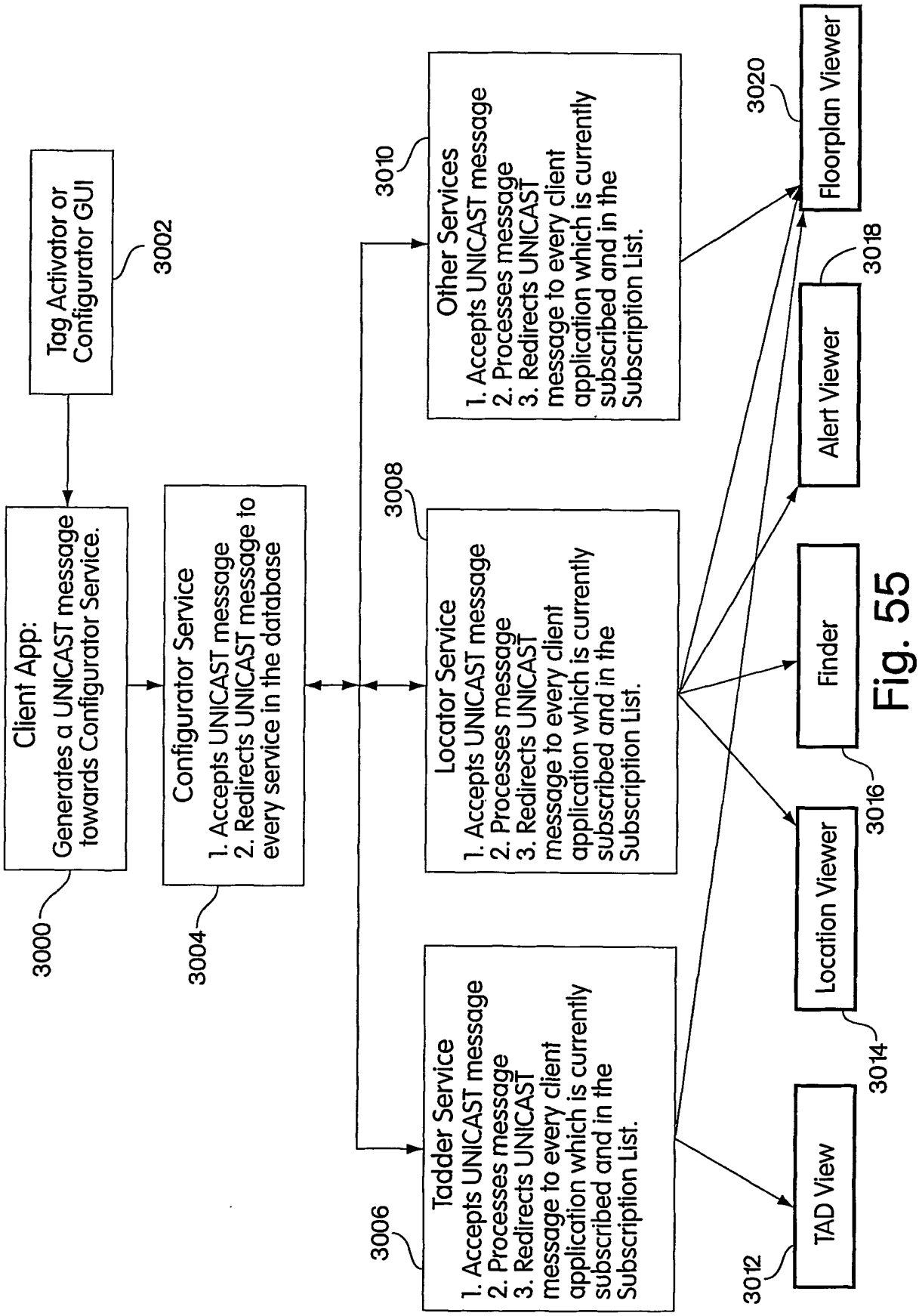

Client App:
Generates a UNICAST message towards Configurator Service.
3000
Tag Activator or Configurator GUI
3002
Configurator Service
1. Accepts UNICAST message
2. Redirects UNICAST message to every service in the database
3004
Tadder Service
1. Accepts UNICAST message
2. Processes message
3. Redirects UNICAST message to every client application which is currently subscribed and in the Subscription List.
3006
Locator Service
1. Accepts UNICAST message
2. Processes message
3. Redirects UNICAST message to every client application which is currently subscribed and in the Subscription List.
3008
Other Services
1. Accepts UNICAST message
2. Processes message
3. Redirects UNICAST message to every client application which is currently subscribed and in the Subscription List.
3010
TAD View
3012
Location Viewer
3014
Finder
3016
Alert Viewer
3018
Floorplan Viewer
3020

Fig. 55

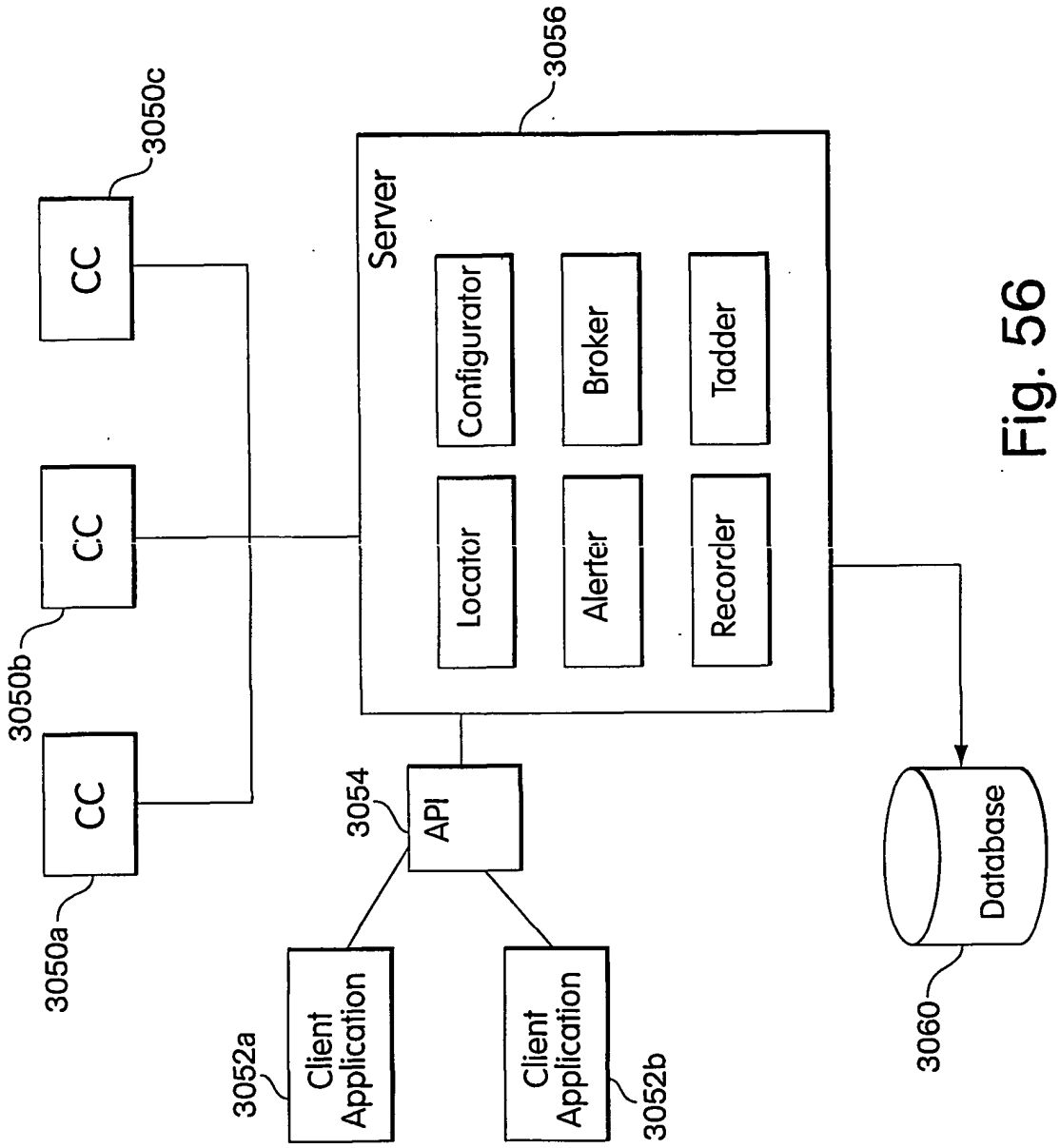

3056
Server
Configurator
Broker
Tadder
Locator
Alerter
Recorder
3050c
CC
3050b
CC
3050a
CC
3054
API
3052a
Client Application
3052b
Client Application
Database
3060

Fig. 56

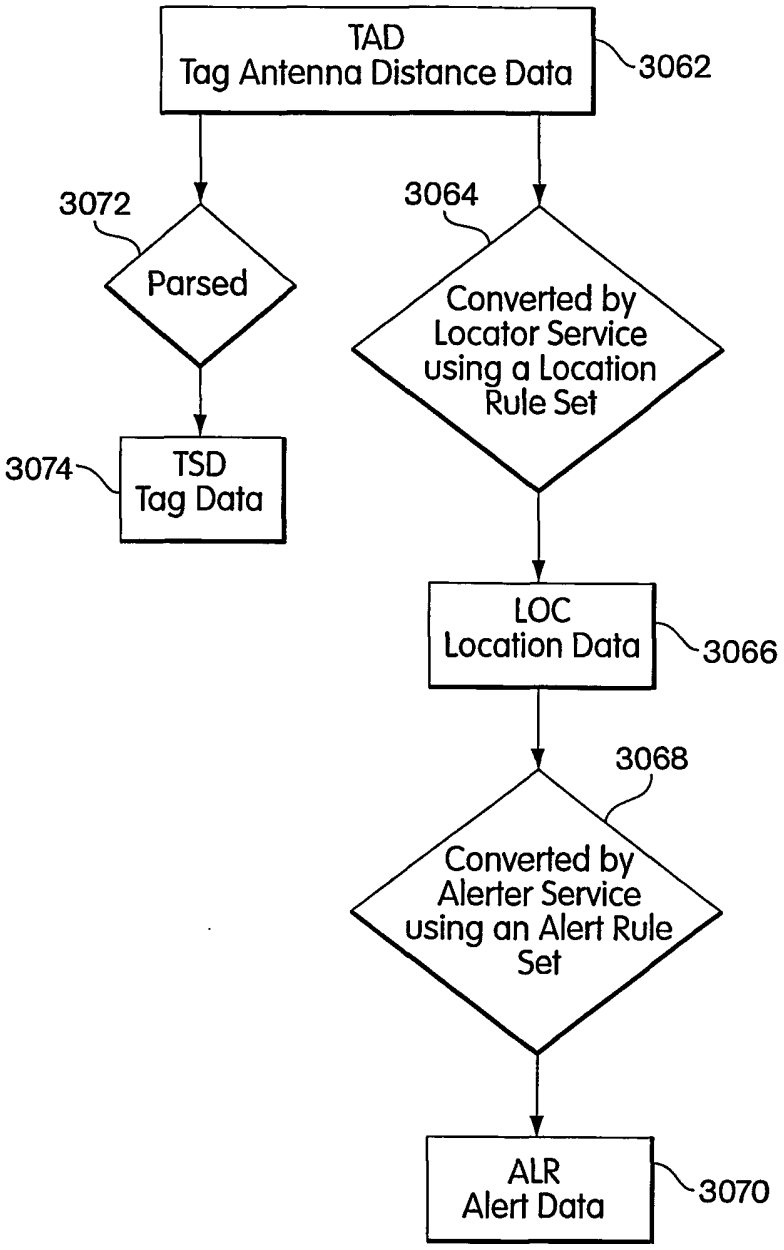

TAD
Tag Antenna Distance Data
3062
3072
Parsed
3064
Converted by Locator Service using a Location Rule Set
3074
TSD
Tag Data
LOC
Location Data
3066
3068
Converted by Alerter Service using an Alert Rule Set
ALR
Alert Data
3070

Fig. 57

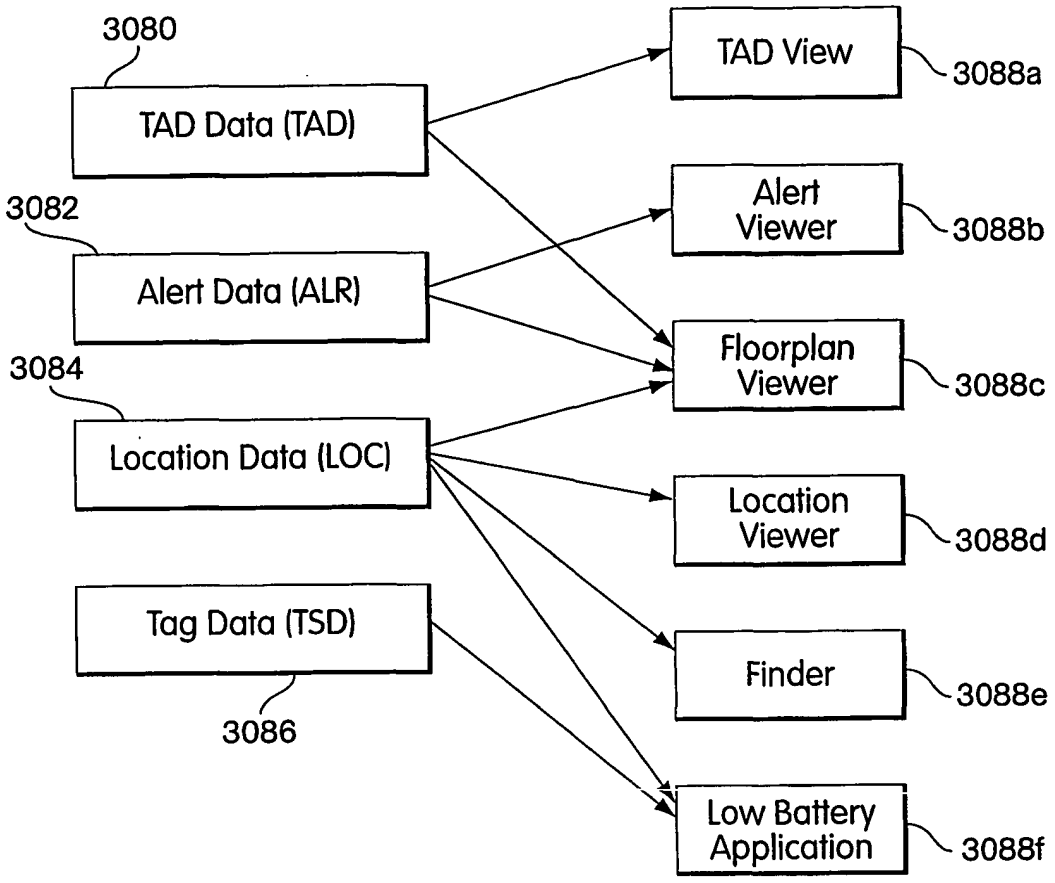

3080
TAD Data (TAD)
3082
Alert Data (ALR)
3084
Location Data (LOC)
Tag Data (TSD)
3086
TAD View
3088a
Alert Viewer
3088b
Floorplan Viewer
3088c
Location Viewer
3088d
Finder
3088e
Low Battery Application
3088f

Fig. 58

## TAD DATA FORMAT

TAD messages adhere to the following format:

```
TAD,<TagID>,<Time>,<# of Antennas>,{<CCID>,<Antenna ID>,<TAD>,...},<Version>,<Status>,<User Data>
```

3090

The following is a sample TAD messages:

```
TAG,31,9938575,2,cc1,2,15,cc2,4,10,1,0,0
```

3092

Breaking this message into parts:

**TAD** — indicates that the message is a TAD message

**31** — indicates that the Tag being reported on is Tag 31

**9938575** — indicates the time at which the LOC reading was taken for the second Tag. This time indication shows that the time of the reading is 9938575 seconds after midnight of January 31, 1970

**2** — indicates that two antennas have 'sighted' the Tag.

**cc1** — indicates theCell Controller for the first antenna which 'sighted' the Tag.

**2** — indicates the Antenna ID for the first antenna.

**15** — indicates the TAD (distance from the first antenna to the Tag) in feet.

**cc2** — indcates the Cell Controller for the second antenna which 'sighted' the Tag.

**4** — indicates the Antenna ID for the second antenna.

3094a

Fig. 59

Location (LOC) messages adhere to the following format: 3100

`LOC ,<LocID>,<Old LocID>,<NumOfTags>{,<TagID>,<Time>}`

The following is a sample LOC message, 3102

`LOC,20,19,2,45,9938575.2,48, 9938576.2`

Breaking this message into parts: 3104

`LOC`—shows that the message is a LOC message.

`20`—indicates the Location the Tag is currently in - further information on the specified Location can be obtained from the Configurator.

`19`—indicates the location the Tag was previously in - further information the specified location can be obtained from the Configurator.

`2`—indicates the number of Tags for which LOC data is being passed.

`45`—is the TagID for the first Tag whose location data is being passed. Information about Tags can be obtained from the Configurator.

`9938575.2`—indicates the time at which the LOC reading was taken for the first Tag. This time indication shows that the time of the reading is 9938575.2 seconds after midnight of January 31,1970. The CvrtTime Control can be used to convert this time to coventional human readable format.

`48`—is the TagID for the second Tag whose location data is being passed. Information about Tags can be obtained from the Configurator.

`9938576.2`—indicates the time at which the LOC reading was taken for the second. Tag. This time indication shows that the time of the reading is 9938575.2 seconds after midnight of January 31,1970.

Fig. 60

An Alert (ALR) message adheres to the following format:
**ALR, <Alert Type>,<ALRID>,<Time>,<Status,<Tag 1>,<Tag2>,<LOCID>** 3106

A sample message might look like:
**ALR,6,19,9938575.2,1,45,67,24** 3108

Breaking this message into parts:

**ALR** — shows that the message is an Alert message.

**6** — indicates the Alert is a foe Alert (two Tags, which should not be in the same Locations, are in the same locations). The numeric Alert Types are:

| | |
|---|---|
| **1** | Entry |
| **2** | Exit |
| **3** | Escort |
| **4** | Friend |
| **5** | Timeout |
| **6** | Foe |
| **8** | Stationary |

**19** — indicates the Alert which has been fired. Information on a specific Alert can be obtained from the Configurator.

**9938575.2** — indicates the time at which the LOC reading was taken for the second Tag. This time indication shows that the time of the reading is 9938575.2 seconds after midnight of January 31, 1970.

**1** — is the state of the alert. 1 means on, 0 means off.

**45** — indicates the first Tag in the Alert. Information on a specific Tag can be obtained from the Configurator

**67** — indicates the second Tag in the alert, if the alert was a single Tag alert (an exit alert for instance),then the Tag 2 value would be identical to the Tag 1 value. Information on a specific Tag can be obtained from the Configurator.

**24** — indicates the location in which the alert was fired. Information on a spacific Tag can be obtained from the Configurator.

3110

Fig. 61

TSD messages are derived from the Tag Data in a TAD message. The Tag Status data is a decimal number from 0 to 255 that represents the combination of Tag Stautus bits. Currently defined status bits are as follows:

Bit 0
Low Battery (0 = battery OK, 1 =battery low)

Bit 1-15
Undefined

The TSD message borrows from the TAD message to produce a message which adheres to the following format:

```
TSD,<Status ID>,<Tag ID>,<State>,<First Date/Time>,<Last Date/Time>
```

3112

A sample message might look like:

```
TSD,1,1932,1,9938575.2,9938586.6
```

3114

Breaking this message into parts:

**TSD** — shows that the message is a Tag Data message.

**1** — indicates that the Tag's battery is low.

**1932** — indicates which Tag has a low battery.

**1** — indicates the state of the bit (0 for cleared, 1 for set).

**9938575.2** — First time notification was received for this status.

**9938586.6** — Last time notification was received for this status.

3116

Fig. 62

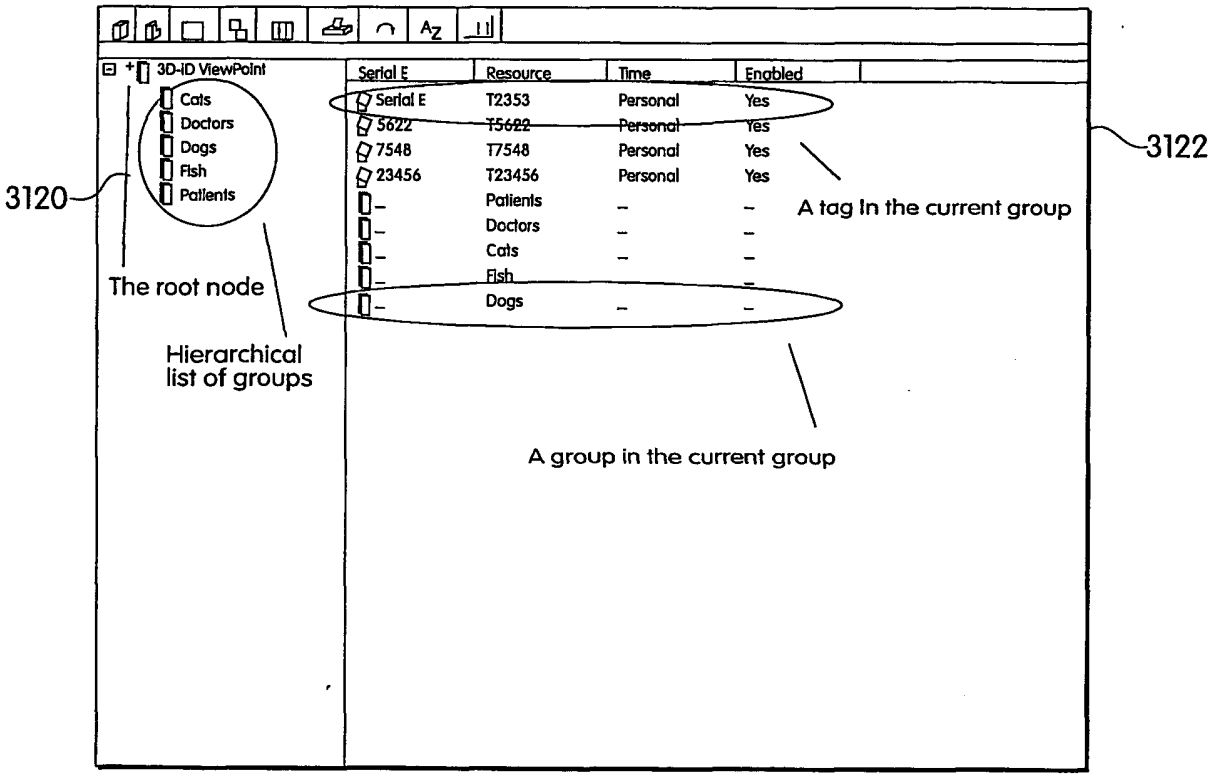

3D-ID ViewPoint
Cats
Doctors
Dogs
Fish
Patients
3120
The root node
Hierarchical list of groups
Serial E
Resource
Time
Enabled
Serial E T2353 Personal Yes
5622 T5622 Personal Yes
7548 T7548 Personal Yes
23456 T23456 Personal Yes
Patients
Doctors
Cats
Fish
Dogs
3122
A tag in the current group
A group in the current group

Fig. 63

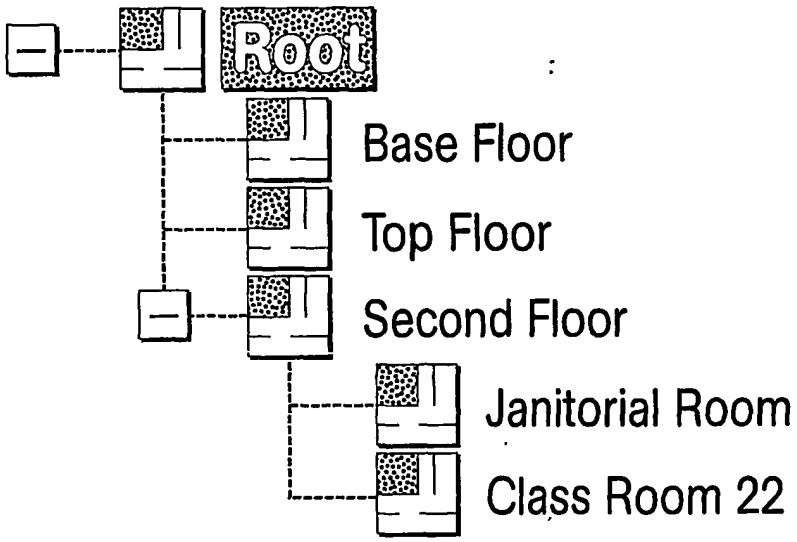

Root
Base Floor
Top Floor
Second Floor
Janitorial Room
Class Room 22

Fig. 64

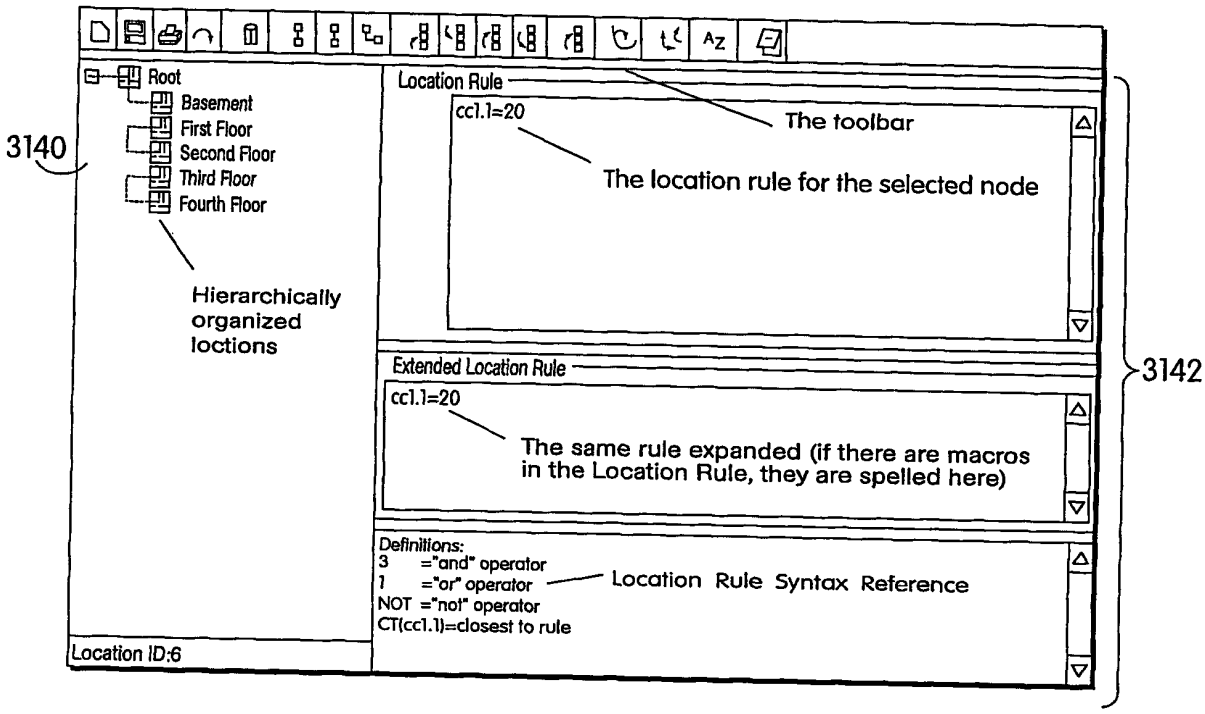

Root
Basement
First Floor
Second Floor
Third Floor
Fourth Floor
3140
Hierarchically organized loctions
Location Rule
cc1.1=20
The toolbar
The location rule for the selected node
Extended Location Rule
cc1.1=20
The same rule expanded (if there are macros in the Location Rule, they are spelled here)
Definitions:
3 ="and" operator
1 ="or" operator
NOT ="not" operator
CT(cc1.1)=closest to rule
Location Rule Syntax Reference
Location ID:6
3142

Fig. 65

Location Rule Syntax

The most basic valid form for a rule, which we will call a simpleform, is the following.

**`CellControllerName.AntennaIndex=[MinRange-]MaxRange`** 3144

(Note: If MinRange is not included in the rule, it defaults to 0.)

An example *simpleform* would be:

**`ccLobby.1=40`**

Which would mean that any tag closer than 40 feet of antenna 1 on the Cell Controller named ccLobby would be included on the node. 3146

The most commonly used *simpleform* is the "closest to " *simpleform*. It takes the form of:

**`CT(ccLobby.1)`**

Which means that any tag which is closest to antenna 1 on the Cell Controller named ccLobby would be included on the node. 3148

Fig. 66

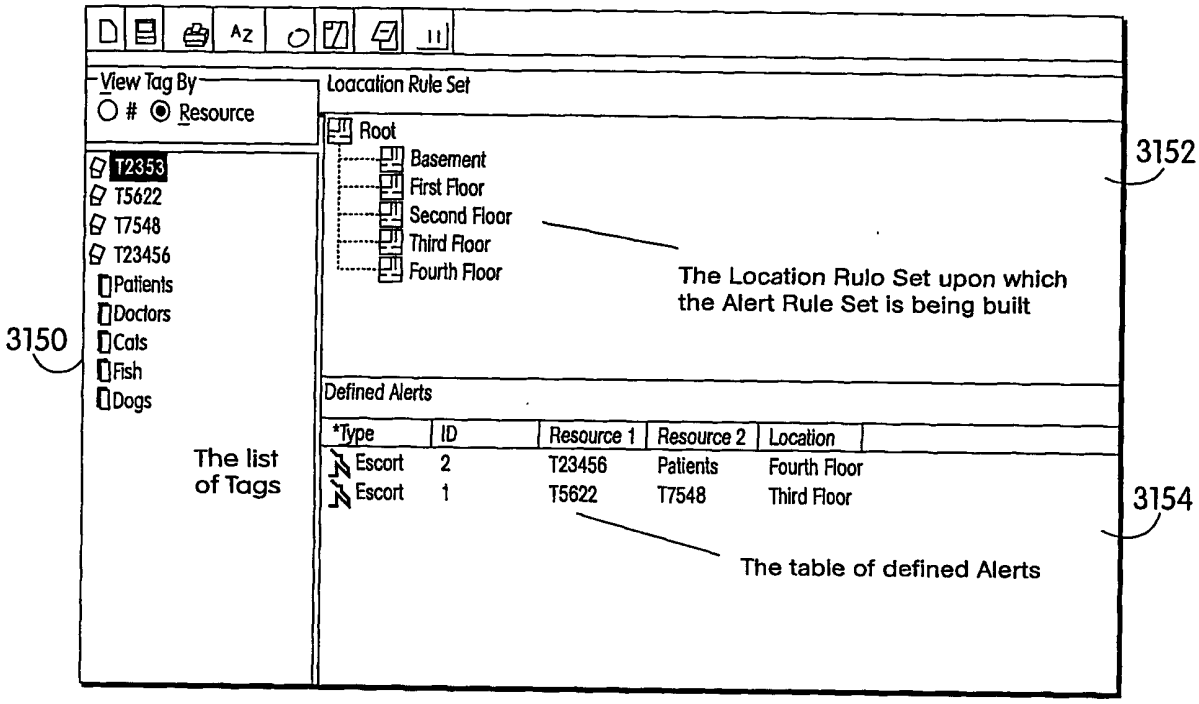

View Tag By
# Resource
Location Rule Set
Root
Basement
First Floor
Second Floor
Third Floor
Fourth Floor
3152
The Location Rule Set upon which the Alert Rule Set is being built
T2353
T5622
T7548
T23456
Patients
Doctors
Cats
Fish
Dogs
3150
The list of Tags
Defined Alerts
*Type
ID
Resource 1
Resource 2
Location
Escort 2 T23456 Patients Fourth Floor
Escort 1 T5622 T7548 Third Floor
3154
The table of defined Alerts

Fig. 67

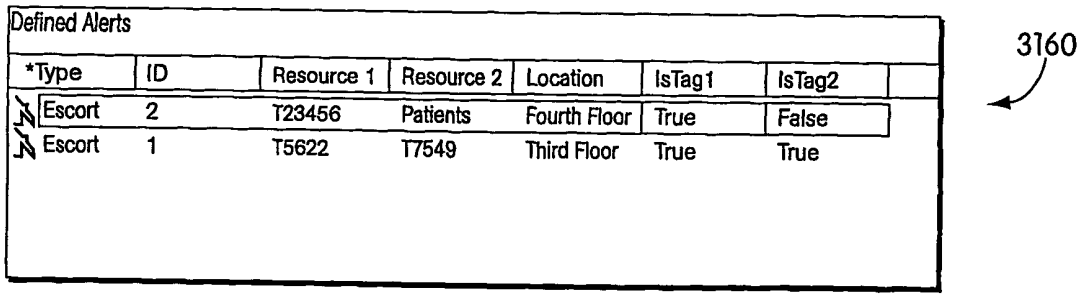

Defined Alerts
*Type
ID
Resource 1
Resource 2
Location
IsTag1
IsTag2
Escort 2 T23456 Patients Fourth Floor True False
Escort 1 T5622 T7549 Third Floor True True
3160

Fig. 68

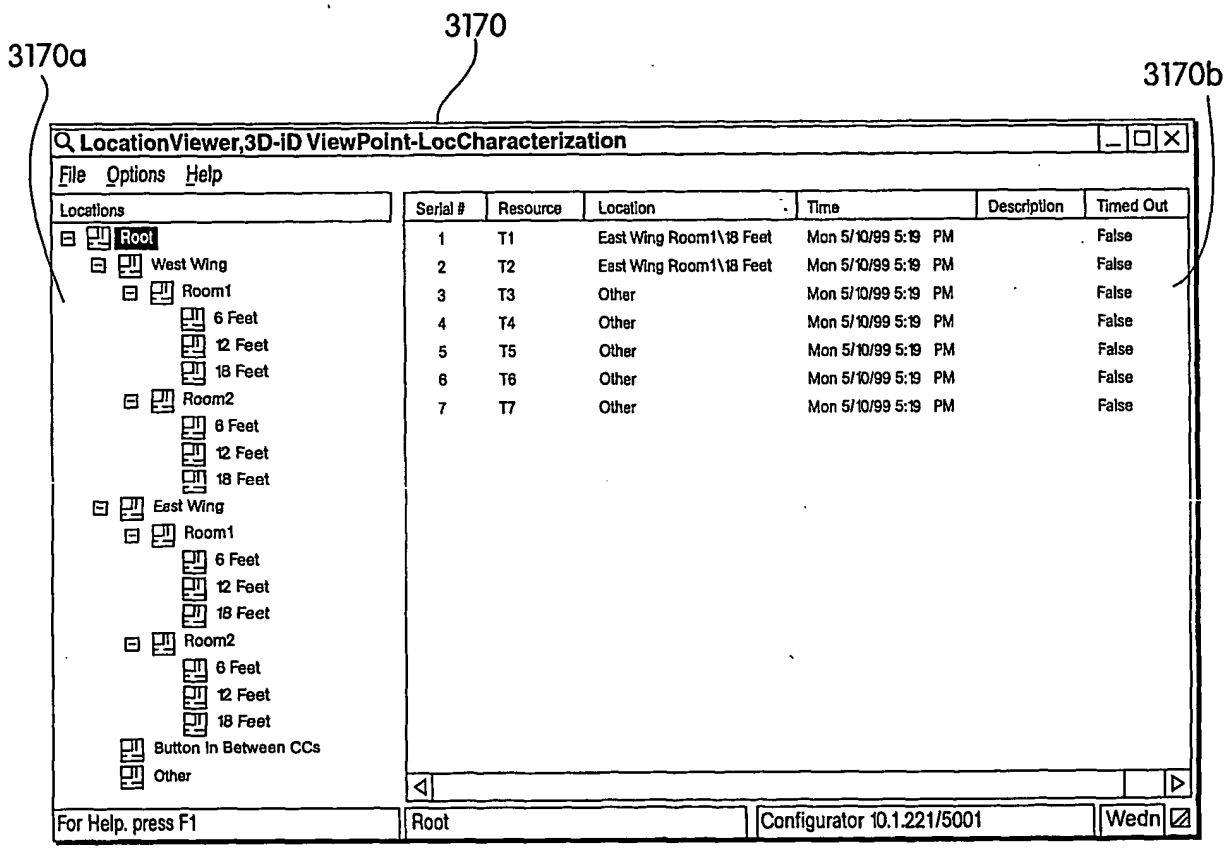

3170
3170a
3170b
LocationViewer,3D-iD ViewPoint-LocCharacterization
File Options Help
Locations
Root
West Wing
Room1
6 Feet
12 Feet
18 Feet
Room2
6 Feet
12 Feet
18 Feet
East Wing
Room1
6 Feet
12 Feet
18 Feet
Room2
6 Feet
12 Feet
18 Feet
Button In Between CCs
Other
Serial # Resource Location Time Description Timed Out
1 T1 East Wing Room1\18 Feet Mon 5/10/99 5:19 PM False
2 T2 East Wing Room1\18 Feet Mon 5/10/99 5:19 PM False
3 T3 Other Mon 5/10/99 5:19 PM False
4 T4 Other Mon 5/10/99 5:19 PM False
5 T5 Other Mon 5/10/99 5:19 PM False
6 T6 Other Mon 5/10/99 5:19 PM False
7 T7 Other Mon 5/10/99 5:19 PM False
For Help, press F1
Root
Configurator 10.1.221/5001
Wedn

Fig. 69

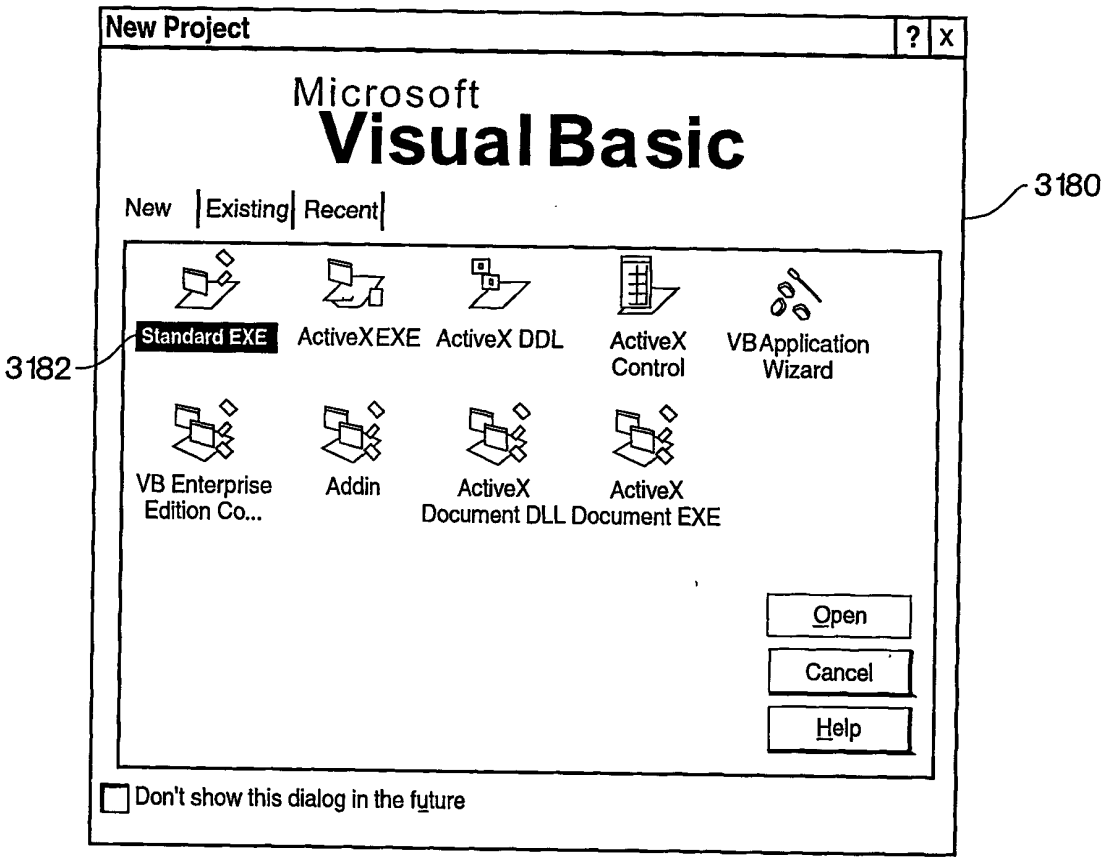

New Project
?
X
Microsoft
Visual Basic
New
Existing
Recent
3180
3182
Standard EXE
ActiveX EXE
ActiveX DDL
ActiveX Control
VB Application Wizard
VB Enterprise Edition Co...
Addin
ActiveX Document DLL
ActiveX Document EXE
Open
Cancel
Help
Don't show this dialog in the future

Fig. 70

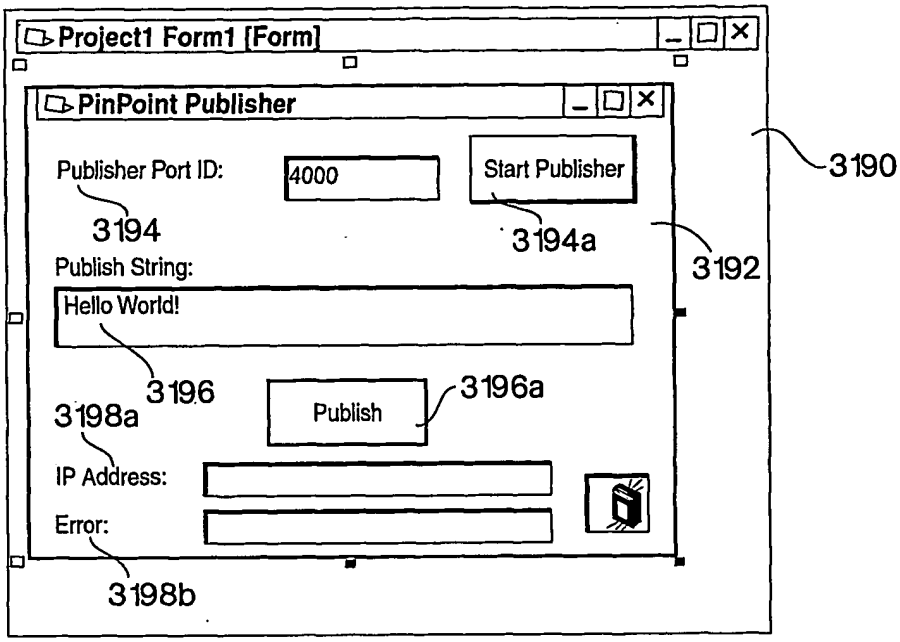



Fig. 71

```
Private Sub cmdPublish_Click()
Dim IRet As Long
IRet = PPPublishX1.PublishMsg(txtString.Text, "TAD")
End Sub
```

3200a 3200b 3200c 3200d 3200

Fig. 72

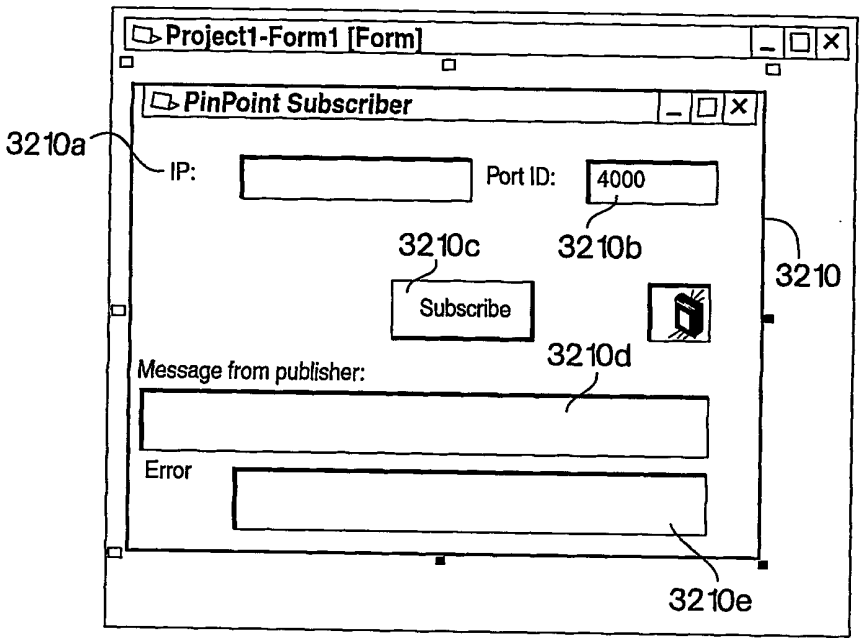

Project1-Form1 [Form]
PinPoint Subscriber
3210a
IP:
Port ID:
4000
3210c
3210b
3210
Subscribe
Message from publisher:
3210d
Error
3210e

Fig. 73

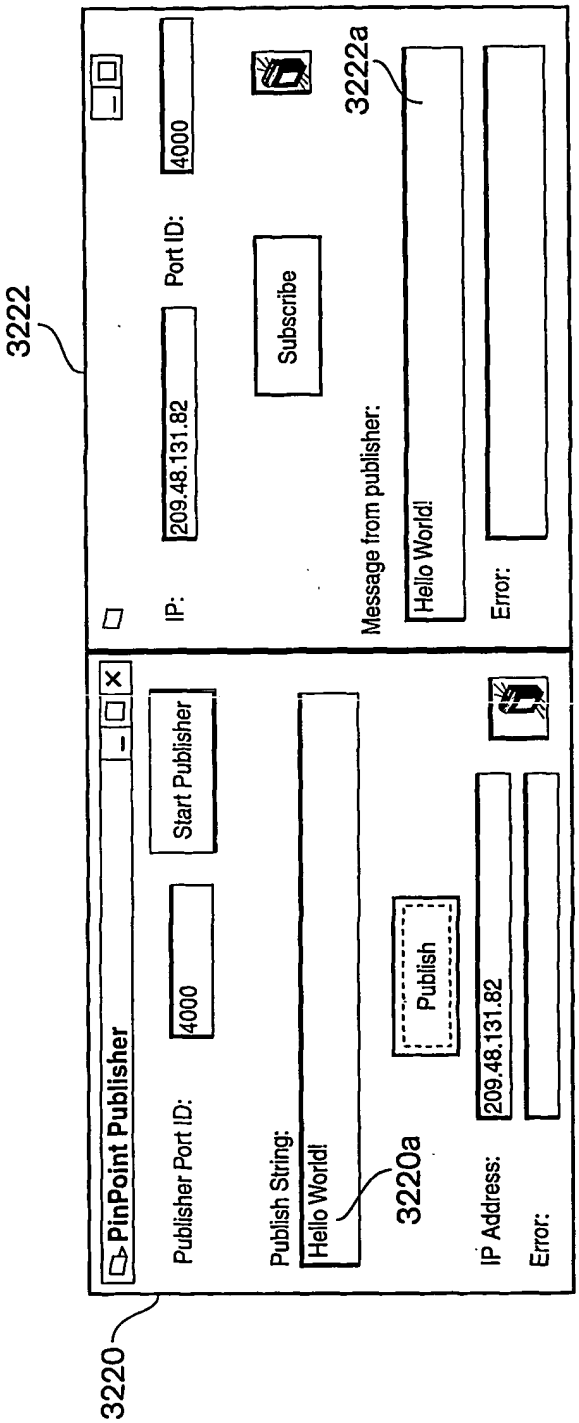

3222
3222a
3220a
3220
PinPoint Publisher
Publisher Port ID:
4000
Start Publisher
Publish String:
Hello World!
Publish
IP Address:
209.48.131.82
Error:
IP:
209.48.131.82
Port ID:
4000
Subscribe
Message from publisher:
Hello World!
Error:

Fig. 74

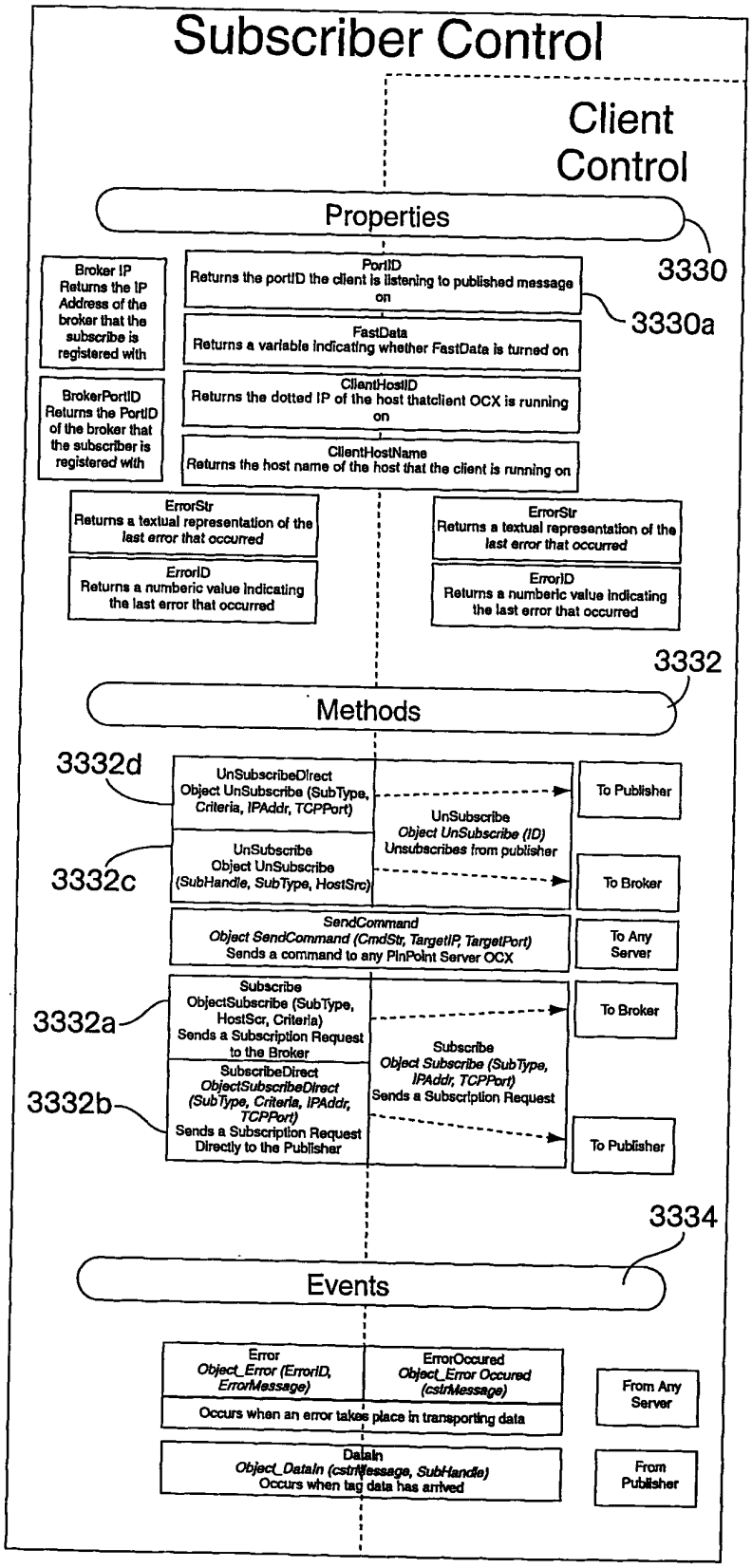

Subscriber Control
Client Control
Properties
3330
Broker IP
Returns the IP Address of the broker that the subscribe is registered with
PortID
Returns the portID the client is listening to published message on
3330a
FastData
Returns a variable indicating whether FastData is turned on
BrokerPortID
Returns the PortID of the broker that the subscriber is registered with
ClientHostID
Returns the dotted IP of the host thatclient OCX is running on
ClientHostName
Returns the host name of the host that the client is running on
ErrorStr
Returns a textual representation of the last error that occurred
ErrorID
Returns a numberic value indicating the last error that occurred
ErrorStr
Returns a textual representation of the last error that occurred
ErrorID
Returns a numberic value indicating the last error that occurred
3332
Methods
3332d
UnSubscribeDirect
Object UnSubscribe (SubType, Criteria, IPAddr, TCPPort)
UnSubscribe
Object UnSubscribe (ID)
Unsubscribes from publisher
To Publisher
3332c
UnSubscribe
Object UnSubscribe (SubHandle, SubType, HostSrc)
To Broker
SendCommand
Object SendCommand (CmdStr, TargetIP, TargetPort)
Sends a command to any PinPoint Server OCX
To Any Server
3332a
Subscribe
ObjectSubscribe (SubType, HostScr, Criteria)
Sends a Subscription Request to the Broker
To Broker
3332b
SubscribeDirect
ObjectSubscribeDirect (SubType, Criteria, IPAddr, TCPPort)
Sends a Subscription Request Directly to the Publisher
Subscribe
Object Subscribe (SubType, IPAddr, TCPPort)
Sends a Subscription Request
To Publisher
3334
Events
Error
Object_Error (ErrorID, ErrorMessage)
ErrorOccured
Object_Error Occured (cstrMessage)
Occurs when an error takes place in transporting data
From Any Server
DataIn
Object_DataIn (cstrMessage, SubHandle)
Occurs when tag data has arrived
From Publisher

Fig. 75

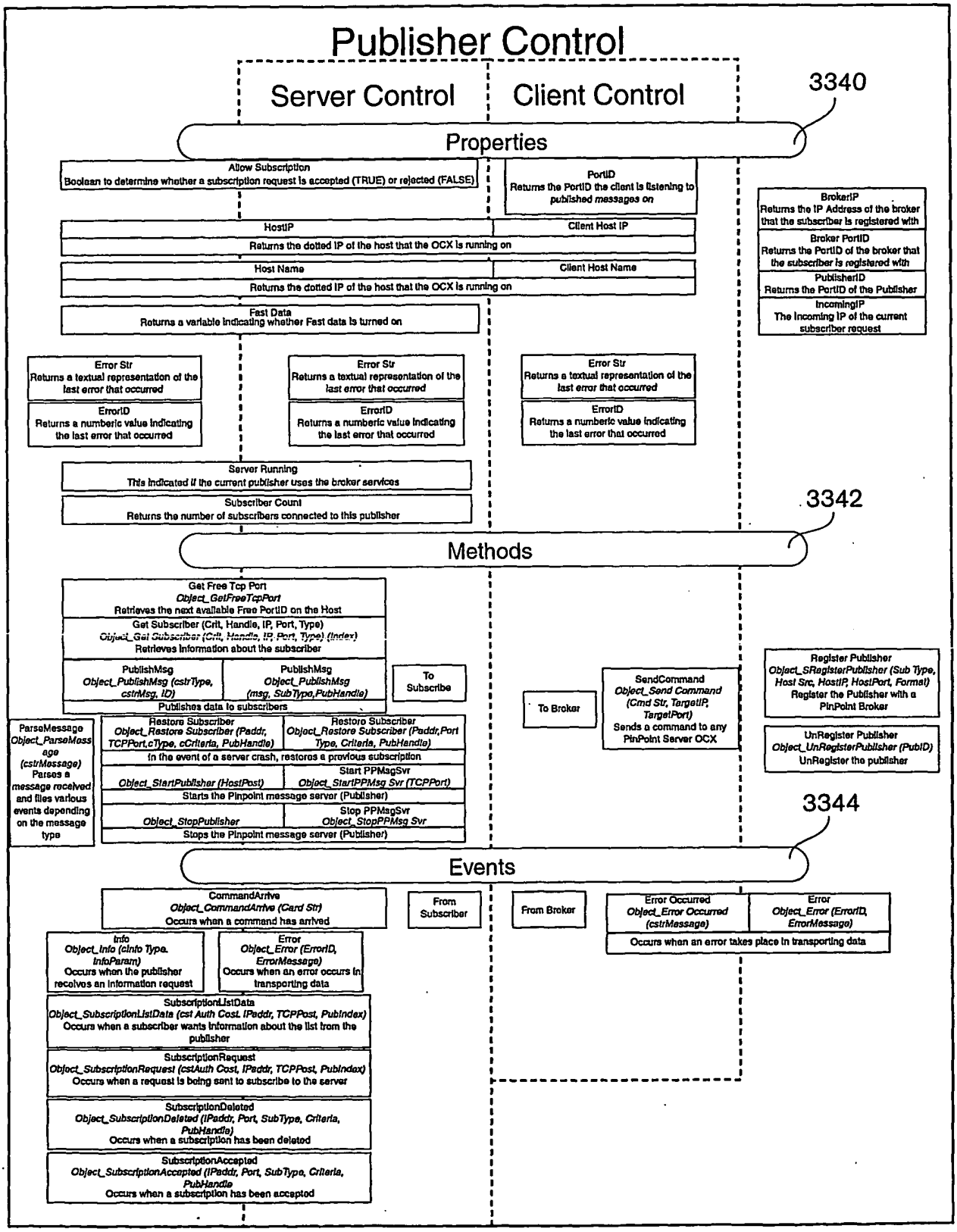

Publisher Control
Server Control
Client Control
3340
Properties
Allow Subscription
Boolean to determine whether a subscription request is accepted (TRUE) or rejected (FALSE)
PortID
Returns the PortID the client is listening to published messages on
BrokerIP
Returns the IP Address of the broker that the subscriber is registered with
Broker PortID
Returns the PortID of the broker that the subscriber is registered with
PublisherID
Returns the PortID of the Publisher
IncomingIP
The Incoming IP of the current subscriber request
HostIP
Client Host IP
Returns the dotted IP of the host that the OCX is running on
Host Name
Client Host Name
Returns the dotted IP of the host that the OCX is running on
Fast Data
Returns a variable indicating whether Fast data is turned on
Error Str
Returns a textual representation of the last error that occurred
ErrorID
Returns a numberic value indicating the last error that occurred
Error Str
Returns a textual representation of the last error that occurred
ErrorID
Returns a numberic value indicating the last error that occurred
Error Str
Returns a textual representation of the last error that occurred
ErrorID
Returns a numberic value indicating the last error that occurred
Server Running
This indicated if the current publisher uses the broker services
Subscriber Count
Returns the number of subscribers connected to this publisher
3342
Methods
Get Free Tcp Port
Object_GetFreeTcpPort
Retrieves the next available Free PortID on the Host
Get Subscriber (Crit, Handle, IP, Port, Type)
Object_Get Subscriber (Crit, Handle, IP, Port, Type) (Index)
Retrieves information about the subscriber
PublishMsg
Object_PublishMsg (cstrType, cstrMsg, ID)
PublishMsg
Object_PublishMsg (msg, SubType,PubHandle)
Publishes data to subscribers
To Subscribe
Register Publisher
Object_SRegisterPublisher (Sub Type, Host Src, HostIP, HostPort, Format)
Register the Publisher with a PinPoint Broker
UnRegister Publisher
Object_UnRegisterPublisher (PubID)
UnRegister the publisher
To Broker
SendCommand
Object_Send Command (Cmd Str, TargetIP, TargetPort)
Sends a command to any PinPoint Server OCX
ParseMessage
Object_ParseMessage (cstrMessage)
Parses a message received and files various events depending on the message type
Restore Subscriber
Object_Restore Subscriber (Paddr, TCPPort,cType, cCriteria, PubHandle)
Restore Subscriber
Object_Restore Subscriber (Paddr,Port Type, Criteria, PubHandle)
In the event of a server crash, restores a previous subscription
Start PPMsgSvr
Object_StartPublisher (HostPost)
Object_StartPPMsg Svr (TCPPort)
Starts the Pinpoint message server (Publisher)
Stop PPMsgSvr
Object_StopPublisher
Object_StopPPMsg Svr
Stops the Pinpoint message server (Publisher)
3344
Events
CommandArrive
Object_CommandArrive (Card Str)
Occurs when a command has arrived
From Subscriber
From Broker
Error Occurred
Object_Error Occurred (cstrMessage)
Error
Object_Error (ErrorID, ErrorMessage)
Occurs when an error takes place in transporting data
Info
Object_Info (cInfo Type, InfoParam)
Occurs when the publisher receives an information request
Error
Object_Error (ErrorID, ErrorMessage)
Occurs when an error occurs in transporting data
SubscriptionListData
Object_SubscriptionListData (cst Auth Cost, IPaddr, TCPPost, PubIndex)
Occurs when a subscriber wants information about the list from the publisher
SubscriptionRequest
Object_SubscriptionRequest (cstAuth Cost, IPaddr, TCPPost, PubIndex)
Occurs when a request is being sent to subscribe to the server
SubscriptionDeleted
Object_SubscriptionDeleted (IPaddr, Port, SubType, Criteria, PubHandle)
Occurs when a subscription has been deleted
SubscriptionAccepted
Object_SubscriptionAccepted (IPaddr, Port, SubType, Criteria, PubHandle
Occurs when a subscription has been accepted

Fig. 76

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/60 G01S13/02

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F H04R G01S

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category ° | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| P,X | US 5 926 108 A (WICKS JAMES E ET AL)<br>20 July 1999 (1999-07-20)<br>abstract; claims 1,2,4,8,9; figures 1-4<br>column 1, line 10 -column 5, line 8 | 1-15 |
| X | US 5 303 393 A (NOREEN GARY ET AL)<br>12 April 1994 (1994-04-12) | 6-14 |
| A | abstract; claims 1-8; figures 1-4<br>column 1, line 50 -column 3, line 20<br>column 4, line 3 - line 47<br>column 6, line 3 - line 40<br>column 11, line 63 -column 15, line 46 | 1-5 |

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 29 December 1999 | 11/01/2000 |

| Name and mailing address of the ISA | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL – 2280 HV Rijswijk<br>Tel. (+31–70) 340–2040, Tx. 31 651 epo nl,<br>Fax: (+31–70) 340–3016 | Gardiner, A |

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

| Category ° | Citation of document, with indication,where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| Y | EP 0 750 200 A (AT & T CORP)<br>27 December 1996 (1996-12-27) | 1-5,15 |
| A | abstract; claims 1-13; figures 1-5,7<br>column 2, line 5 -column 7, line 19 | 6-14 |
| Y | US 5 337 044 A (ARNOLD DAVID ET AL)<br>9 August 1994 (1994-08-09) | 1-5,15 |
| A | abstract; claims 1-4,12-14; figure 1<br>column 1, line 21 -column 4, line 57<br>column 5, line 19 -column 6, line 1<br>column 8, line 42 -column 9, line 46 | 6-14 |
| A | US 4 746 830 A (HOLLAND WILLIAM R)<br>24 May 1988 (1988-05-24)<br>abstract; claims 1-11; figures 1-4<br>column 1, line 6 -column 5, line 11 | 1,6,7,<br>11,15 |

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 5926108 A | 20-07-1999 | NONE | |
| US 5303393 A | 12-04-1994 | US 5455823 A | 03-10-1995 |
| EP 0750200 A | 27-12-1996 | US 5649295 A | 15-07-1997 |
| | | CA 2177415 A | 20-12-1996 |
| | | JP 9027773 A | 28-01-1997 |
| US 5337044 A | 09-08-1994 | NONE | |
| US 4746830 A | 24-05-1988 | NONE | |